МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут інноваційних освітніх технологій
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____Савченко А.С.
“ ”_____2020 р.

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ

“МАГІСТР”

Тема: “Технологія розробки шаблонів для платформи Wordpress”

Виконав: Дзвонкевич Савелій Антонович

Керівник: Моржов В. І.

Нормоконтролер з ЄСКД (ЄСПД): Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕСИТЕТ

Навчально-науковий інститут заочного та дистанційного начання

Кафедра комп’ютерних інформаційних технологій

Освітньо-кваліфікаційний рівень: Магістр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп’ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____

“ ” _____ 2020р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Дзвонкевич Савелій Антонович

(прізвище, ім'я, по батькові)

1. Тема дипломного проекту: «Технологія розробки шаблонів для платформи WordPress» затверджена наказом ректора від 18.01.2020 № 10/ст

2. Термін виконання проекту: з 18.01.2020р. до 25.02.2020

3. Вихідні данні до проекту: шаблон за допомогою якого можливо створювати веб-сайти із унікальним дизайном та зрозумілим інтерфейсом адміністрування контенту.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

Типи веб-сайтів, парадігми проектування, платформи для автоматизації розробки, робоча середа розробки шаблонів, технології для роботи із

статичними та динамічними файлами шаблонів, платформи для шаблонізації хмарного зберігання коду.

5. Перелік обов'язкового графічного матеріалу: слайди Power Point.

Календарний план-графік

№ пор	Завдання	Термін виконання	Відмітка про виконання
1.	Виконати детальний аналіз літератури та Інтернет джерел за темою дипломного проекту. Розробити план дипломного проекту.		
2.	Затвердити план дипломного проекту.Написати та затвердити вступ до дипломного проекту Провести консультації з науковим керівником, щодо першого розділу.		
3.	Загальний огляд методів створення веб-порталів		
4.	Аналіз та порівняння CMS, локальний сервер Denwer		
5.	Створення веб-порталу інтернет-магазину на базі CMS «WordPress»		
6.	Написати висновки до виконаного дипломного проекту.Оформити пояснювальну записку.Підготовити доповідь та презентацію.		
7	Підписати необхідні документи у встановленому порядку. Підготовитися до захисту дипломного проекту.		

7. Дата видачі завдання: « 18 » 01 2020 р.

Керівник дипломного проекту _____ Моржов В. І.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Дзвонкевич С. А.

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Технологія розробки шаблонів для платформи WordPress»: 104 сторінки, 39 рисунків, 16 використаних джерел.

Ключові слова: АВТОМАТИЗАЦІЯ, ШАБЛОН (ТЕМА).

Об'єкт дослідження – Шаблон на базі платформи Wordpress.

Предмет дослідження – Технологія автоматизації проектування та розробки веб-сайтів.

Мета проекту – полягає у автоматизації процесу проектування та розробки веб-сайтів за допомогою шаблонів на різних етапах розробки.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

CMS – Content Management System (Система керування контентом)

SQL (англ. Structured query language — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних.

AJAX (Asynchronous JavaScript And XML) — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись.

WWW – World Wide Web (Всесвітня мережа)

XHTML – Extensible hypertext markup language (Розширювана мова розмітки гіпертексту)

API – Application Programming Interface (Прикладний програмний інтерфейс)

FTP - Протокол передачі файлів (англ. File Transfer Protocol, FTP).

XSS (англ. Cross Site Scripting) — «міжсайтовий скриптинг»

JPG (JPEG) - Joint Photographic Experts Group

PNG (Portable Network Graphics) — растровий формат збереження графічної інформації

PHP – Hypertext Preprocessor (Гіпертекстовий препроцесор)

ОС – Операційна система

JSON (англ. JavaScript Object Notation) — це текстовий формат обміну даними між комп'ютерами.

HTTP – HyperText Transfer Protocol (Протокол передачі гіпер-текстових документів)

URL – Universal Resource Locator (Уніфікований локатор ресурсів)

SEO – Search engine optimization (Пошукова оптимізація сайту)

GIT - система розподіленої версії.

.GITIGNORE - файл вказує навмисно не відстежувані файли, які Git слід ігнорувати.

WYSIWYG – What You See Is What You Get (Що бачиш, те й отримуєш)

.HTACCESS - це локальний конфігураційний файл вебсервера Apache.

RSS (Rich Site Summary) - тип веб-каналу, який дозволяє користувачам отримувати доступ до оновлень онлайн-вмісту..

ВСТУП

З кожним роком можливості веб-ресурсів грають більшу роль у туристичному бізнесі. У згаданнму сегменті істунє ріноманітність пропозицій по всюму світу, важко порахувати кількість готелей хочаб серед курортних регіонів Закарпаття, круїзних подорожей у Середньому та Карибському морях, гірнолижних курортів – це лише кілька прикладів, але абсолютно всі вони потребують важілей для залучення нових відвідувачів, і одним із таких виявляється створення власного веб-ресурсу.

Вихідні ресурси наведених вище прикладів будуть дуже відрізнятись, наприклад, сайт круїзних подорожей обовьязково має містити детальну інформацію про маршрути своїх круізів, характеристики лайнерів та інформацію міст де лайнер планує робити зупинку, на від міну від сайту гірнолижного курорту, де потрібен інакший функционал.

Маючи на меті автоматизацію процесу проектування та розробки веб-ресурсів, нам необхідно вишикувати етапи розробки кожного нового проекту таким чином щоб вони відбувались по єдиній схемі, розглядаючи кожен етап окремо, маємо можливість залучити інструменти для його автоматизації.

Зазначемо етапи розробки веб-сайту: проектування, программування, та наповнення контентом.

За допомогою платформ для планування проектів можливо автоматизувати процеси проектування, відстеження та керування розробкою. Тож ми розглянемо сучасні парадігми проектування, та визначимо найбільш продуктивний метод.

Також існують спеціальні фрейм-ворки які містять увесь необхідний программний код для функціювання сайту та лише потребують від користувача наповнення сайту інформацією без втручання у код. Наприклад, такі фрейм-ворки к Wordpress, Prestashop та Drupal дозволяють обрати тему оформлення (інакшими словами - шаблон) на свій смак за заповнити її власним контентом.

Існує два типи тем оформлення: «безкоштовна тема» оформлення с дуже замалими можливостями функціоналу (декілька дуже простих сторінок і можливість створення блогу) та «преміум тема» яка коштує грошей та може включати в себе дуже великий діапазон можливостей, навіть конструктор сторінок. Середня ціна на ліцензію «преміум шаблону» складає $70, але не рідко преміум шаблони працюють не достатньо коректно.

Тож у данній роботі ми розглянемо автоматизацію розробки веб-сайтів за власного шаблону на базі фрейм-ворку Wordpress для туристичного сегменту.

РОЗДІЛ 1: МЕТОДИ РОЗРОБКИ ВЕБ-САЙТІВ

1.1. Типи сайтів

Працюючи над проектуванням веб-сайту необхідно визначити метод розробки, який залежить від типу та цілей веб-застосунку, які за його допомогою повинен досягти власник. Офіційної класифікації сторінок в Інтернеті не існує, але є основні характеристики, за якими можна розділити існуючі сайти.

Розглянемо умовні типи сайтів:

- Сайт-візитка;
- Сайт-вітрина;
- Інтернет-магазин;
- Веб-застосунок.

Сайт-візитка

Саме той набір інформації, поширення якої в Інтернеті, безсумнівно, корисно як починаючим, так і акулам бізнесу. Це своєрідний довідник про фірму, який містить всі необхідні контактні дані та інформацію про діяльність компанії.

Основні переваги:

- зручний засіб для сканування грунту ринку;
- доволі низька ціна на розробку;
- не потребує значних часових ресурсів для розробки.

Корпоративний сайт

За допомогою корпоративного сайту клієнти та замовники отримують інформацію про ціни на товари і послуги в зручному вигляді. Такий сайт може мати вбудований блог с найактуальнішими новинами компанії а також можливістю замовити підписку на новини через пошту.

Сайт-вітрина

Бюджетним варіантом реклами будь-якої компанії є сайт-вітрина. Коли є необхідність продавати в Інтернеті один конкретний товар або групу товарів, то цей тип сайтів беззаперечно підходить для компанії. На таких сторінках знаходиться уся інформація, необхідна для того, щоб клієнт побачив, зацікавився і купив даний товар, не відволікаючись на новини та посилання. Це структурований каталог продукції Вашої компанії. Але цей каталог організований таким чином (фото, опис, ціна), щоб клієнт зробив як можна менше кліків, для того, щоб знайти потрібний товар. Головне призначення сайту-вітрини - продавати.

Інтернет-магазин

На відміну від сайту-вітрини, де клієнт може побачити наявність всіх товарів, за допомогою інтернет-магазину клієнт може ще й зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату. Такий сайт має бути зручним і функціональним, щоб відвідувач міг легко знайти те, що йому потрібно, відправити товар у кошик та оформити покупку в кілька кліків. Інтернет-магазин істотно зменшить Ваші витрати, так, як не треба орендувати приміщення для магазину, не треба платити заробітну плату співробітникам. Серед інших переваг: величезна аудиторія, можливість

розміщувати необмежений асортимент товарів, своєчасно реагувати на зміни на ринку, враховувати потреби клієнтів та інші.

Веб-застосунки

До веб-застовунків можна віднести такі ресурси які поєднуюсь у собі складну функціонльність та широку цільову аудиторію. До веб-застосунків можна віднести:

- Поштові сервіси;
- Сервіси замовлення таксі;
- Пошукові системи;
- Соціальні мережі.

Незважаючи на те, що веб-сайти розрізнюються за типами, при проектуванні веб-сайту необхідно побудувати архитектуру таким чином, щоб було можливо змінити тип сайту у майбутньому.

Наприклад, популярною практикою є розробкка сайту-візитки, та подальшого його транстформування у сайт-вітрину, чи інтернет магазин.

1.2. CMS Wordpress

WordPress — це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення веб-сайтів. Сфера застосування — від блогів до складних веб-сайтів. Вбудована система тем і плагінів в поєднанні з вдалою

архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти.

Написана мовою програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License.

Одне встановлення WordPress дозволяє вести одночасно лише один блог. Якщо ж ви бажаєте створити і вести на одному сервері декілька блогів, то можна встановити потрібну кількість WordPress в різні каталоги (віртуальні сервери) і в СКБД створити окрему базу даних для кожного блогу.

Гілка (англ. fork) WordPress Multi-User (WordPress MU, чи WPMU) дає змогу створити декілька блогів в одній інсталяції. WordPress MU також дозволяє кожному користувачу системи вести довільну кількість блогів і контролювати їх з панелі управління. Для кожного блогу створюється 8 таблиць в базі даних.

На рис. 1 зображена доля сайтів побудованних на системі Вордпресс.

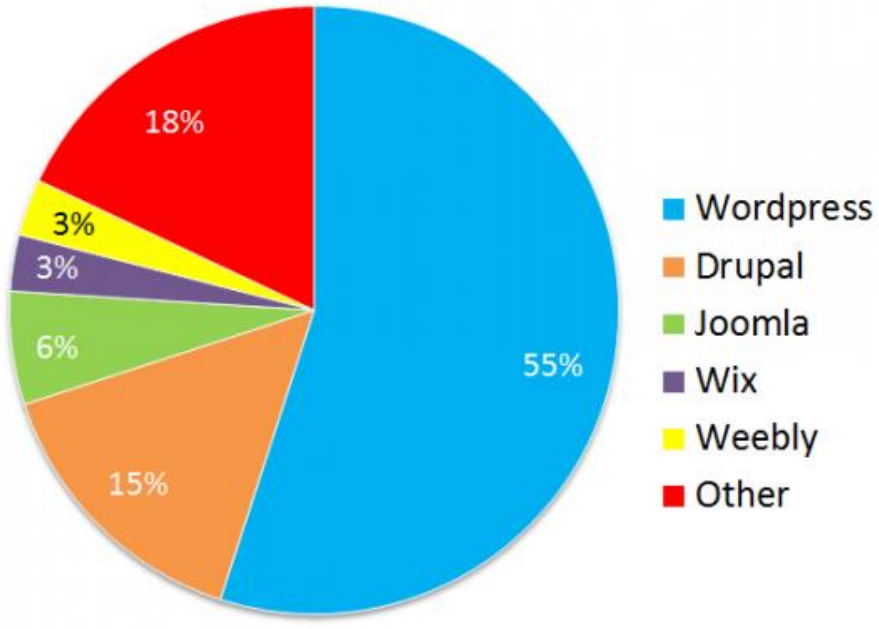


Рис. 1

Ключові можливості

Дизайн та управління системою.

- підтримка веб-стандартів (XHTML, CSS);
- простота встановлення, простота налаштувань;
- модулі для підключення (плаґіни) з унікально простою системою їх взаємодії з кодом;
- можливість автоматичного встановлення та оновлення версії безпосередньо з панелі адміністратора;
- підтримка так званих «тем», з допомогою яких легко змінюється як зовнішній вигляд, так і способи виведення даних;
- можливість редагувати шаблони одразу в панелі адміністратора;
- «теми» реалізовані як набори файлів-шаблонів на PHP (у HTML-розмітку вставляються PHP-мітки);
- багато бібліотек «тем» і «плаґінів»;
- потенціал архітектури дозволяє легко реалізовувати складні рішення;
- SEO-оптимізована система;
- наявність українського перекладу.

Публікація та редагування

- миттєва публікація;
- підтримка RSS, Atom, trackback, pingback;
- наявність ЛЗУ (людино-зрозумілий URL);
- редагування WYSIWYG-редактором з можливістю вставлення форматованого тексту (наприклад з програми Microsoft Word) або редагування за допомогою HTML-розмітки.

Контент

- багатосторінкові записи;
- наперед заплановані публікації;
- прикріплення файлів та зображень до записів;
- можливість створення статичних сторінок;
- можливість створення свого типу контенту у власних темах;
- категорії, теги, коментування тощо.

Структура файлів у корні проекту

Кореневий каталог WordPress містить три папки: wp-content, wp-includes і wp-admin разом з купою різних PHP файлів, які потрібні для основних операцій WP. Найбільш значущим з цих файлів є «wp-config.php». Змінюючи цей файл, можна додати купу ключових варіантів налаштувань WordPress, які не доступні з консолі адміністратора. Також в корені сайту лежать інші системні файли (наприклад, wp-settings.php, wp-config.php), які впливають на налаштування сайту.

Розглянемо коротко анатомію двигуна Вордпрес і за що відповідають ті, чи інші файли та папки.

- wp-admin -- ця папка містить різні файли, такі як CSS, JavaScript і PHP, які забезпечують функціональність консолі та адміністративної частини сайту.
- wp-content - містить всі завантажені дані користувача та розділяється на інші папки:
    - language - містить файли перекладів та локалізації двигуна у форматі .mo та .po.
    - plugins – містить додаткові плагіни

    - themes - містить всі завантажені теми (шаблони)
    - uploads - усі зображення (та інші медіа-файли) зберігаються в каталозі «uploads», з розбивкою по роках, місяцях та/або днях. Ця папка являє собою базу даних для всього не-текстового контенту: зображення, відео, MP3, PDF-файли, і т.д. Відразу після встановлення WordPress папки «uploads» не буде, вона буде створена автоматично, після того як ви почнете завантажувати медіа-файли через консоль.
- wp-includes - містить в собі всі основні та необхідні файли для запуску WordPress через фронтенд (користувальницький інтерфейс). Папка містить файли PHP, CSS, JavaScript та файли зображень WordPress, які забезпечують основні функції програмного забезпечення. Іншими словами – це ядро двигуна Вордпрес.

Структура бази данних

- wp_options - зберігає усі налаштунки сайту, назву, опис, часовий пояз, мову інтерфейсу тощо;
- wp_users – зверігая список усіх зареєстрованних користувачів, а також: логін, пароль зашифрованний у MD5, Роль,Ключ активації, якщо потрібен;
- wp_usermeta – зберігає метаданні користувачів. Наприклад, прізвище зберігається саме у цій таблиці. У цій таблиці є два дуще важливих поля -- meta_key та meta_value. Різноманітні додатки та блигіни можуть використати ці поля для зберігання додаткової інформації про користувачів;
- wp_posts – зберігаєусі данні для:
    - Статичні веб сторінки;
    - Статті блогу;

    - Ревізії;
    - Пункти навігаційних меню;
    - Медіа файли.
- wp_postmeta - зберігая мета-данні вище перелічених записів. Під мета-данними мається на увазі
    - Додаткова інформація;
    - CEO заголовки та описи.
- wp_terms – зберігає категорії, теги для записів та сторінок;
- wp_term_relationship – це посилання на обєкти статей, сторінок, лінків;
- wp_term_taxonomy – таблиц поєднує категорії, лінки, теги та дає змогу робити вложені категорії;
- wp_comments та wp_commentmeta – зберігає коментарії користувачів до статей та сторінок, а також інформацію про автора комментарію.

На Рис. 2 представленна реляційна схема бази данних Вордпресс.

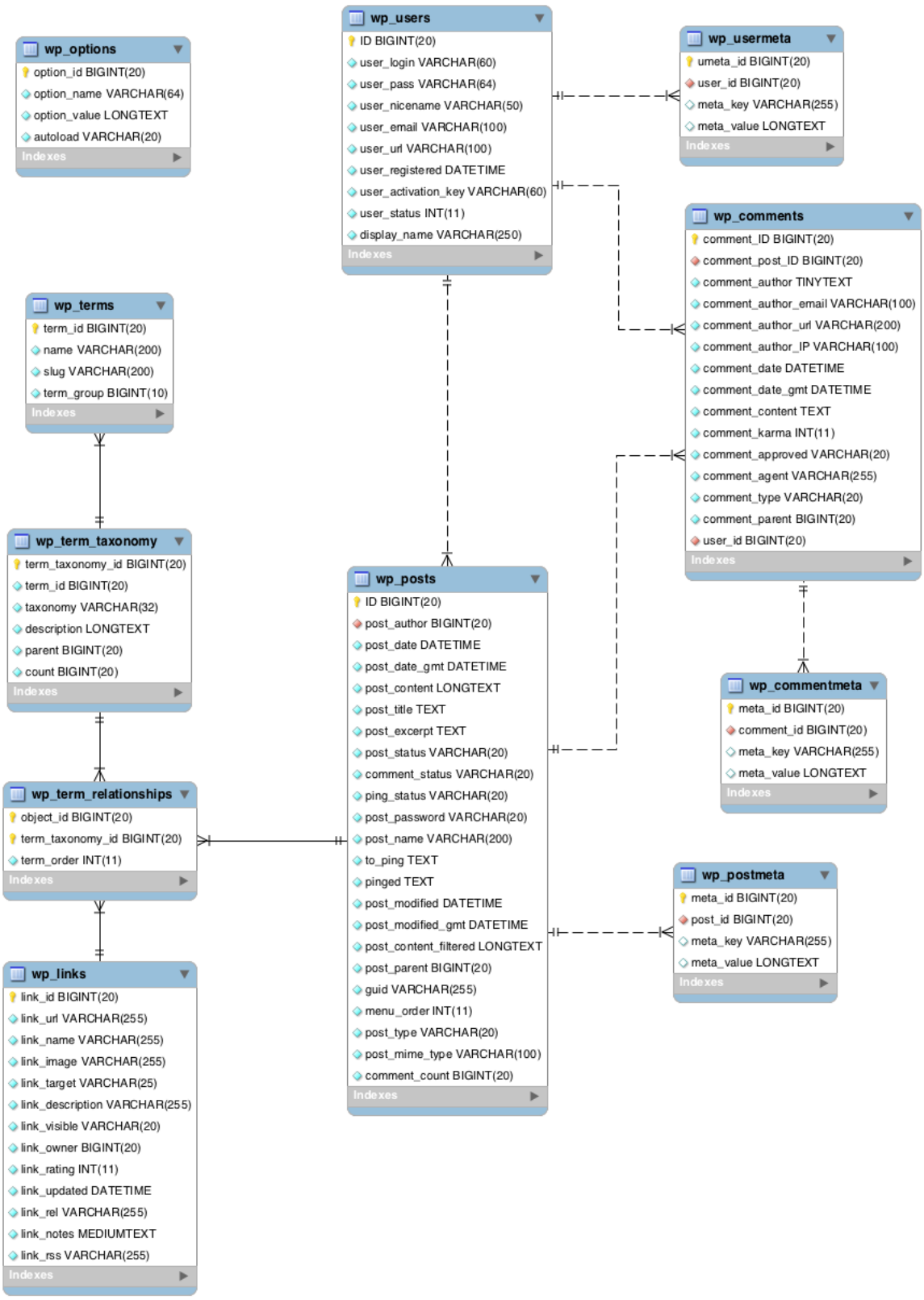

wp_options
option_id BIGINT(20)
option_name VARCHAR(64)
option_value LONGTEXT
autoload VARCHAR(20)
Indexes
wp_users
ID BIGINT(20)
user_login VARCHAR(60)
user_pass VARCHAR(64)
user_nicename VARCHAR(50)
user_email VARCHAR(100)
user_url VARCHAR(100)
user_registered DATETIME
user_activation_key VARCHAR(60)
user_status INT(11)
display_name VARCHAR(250)
Indexes
wp_usermeta
umeta_id BIGINT(20)
user_id BIGINT(20)
meta_key VARCHAR(255)
meta_value LONGTEXT
Indexes
wp_comments
comment_ID BIGINT(20)
comment_post_ID BIGINT(20)
comment_author TINYTEXT
comment_author_email VARCHAR(100)
comment_author_url VARCHAR(200)
comment_author_IP VARCHAR(100)
comment_date DATETIME
comment_date_gmt DATETIME
comment_content TEXT
comment_karma INT(11)
comment_approved VARCHAR(20)
comment_agent VARCHAR(255)
comment_type VARCHAR(20)
comment_parent BIGINT(20)
user_id BIGINT(20)
Indexes
wp_terms
term_id BIGINT(20)
name VARCHAR(200)
slug VARCHAR(200)
term_group BIGINT(10)
Indexes
wp_term_taxonomy
term_taxonomy_id BIGINT(20)
term_id BIGINT(20)
taxonomy VARCHAR(32)
description LONGTEXT
parent BIGINT(20)
count BIGINT(20)
Indexes
wp_posts
ID BIGINT(20)
post_author BIGINT(20)
post_date DATETIME
post_date_gmt DATETIME
post_content LONGTEXT
post_title TEXT
post_excerpt TEXT
post_status VARCHAR(20)
comment_status VARCHAR(20)
ping_status VARCHAR(20)
post_password VARCHAR(20)
post_name VARCHAR(200)
to_ping TEXT
pinged TEXT
post_modified DATETIME
post_modified_gmt DATETIME
post_content_filtered LONGTEXT
post_parent BIGINT(20)
guid VARCHAR(255)
menu_order INT(11)
post_type VARCHAR(20)
post_mime_type VARCHAR(100)
comment_count BIGINT(20)
Indexes
wp_commentmeta
meta_id BIGINT(20)
comment_id BIGINT(20)
meta_key VARCHAR(255)
meta_value LONGTEXT
Indexes
wp_term_relationships
object_id BIGINT(20)
term_taxonomy_id BIGINT(20)
term_order INT(11)
Indexes
wp_postmeta
meta_id BIGINT(20)
post_id BIGINT(20)
meta_key VARCHAR(255)
meta_value LONGTEXT
Indexes
wp_links
link_id BIGINT(20)
link_url VARCHAR(255)
link_name VARCHAR(255)
link_image VARCHAR(255)
link_target VARCHAR(25)
link_description VARCHAR(255)
link_visible VARCHAR(20)
link_owner BIGINT(20)
link_rating INT(11)
link_updated DATETIME
link_rel VARCHAR(255)
link_notes MEDIUMTEXT
link_rss VARCHAR(255)
Indexes

Рис. 2

«Теми» у Вордпрессі

Перш за все Тема Wordpress (або шаблон теми) – це набір файлів, які відповідають за зовнішній вигляд веб-ресурсу у браузері користувача. Тема - це костюм чи одяг для веб-сайту. Основна ідея полягає у тому, что користувач має змогу з легкістю міняти цей одяг (зовнішній вигляд сайту) але при цьому основні функції залишаються не змінними.

Поняття шаблону – це більш вузьке поняття. Шаблон – це набір файлів теми, які формують відображення інформації належним чином. Наприклад шаблон для відображення сторінок блогу, або шаблон для відображення категорій товарів. Шаблони відповідають за те, яку інформацію та у яких місцях її відібражати. Вони пишуться на мові PHP з використанням HTML розмітки.

Система Вордпрессу надає змогу безкоштовно завантажити тему через панель адміністратора (рис. 3)

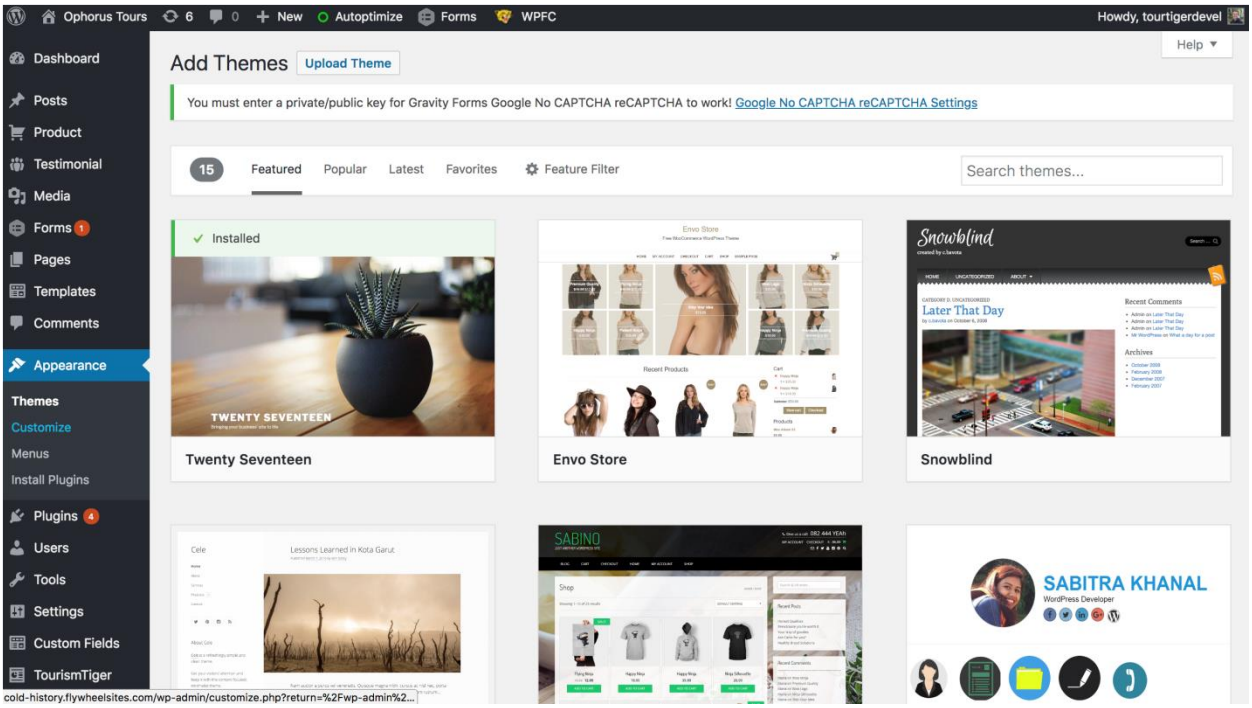


Рис. 3

Основна структура папок теми може включати в себе такі папки, як:

- /css/ – файли стилів CSS, які використовує тема. ВАЖЛИВО: файл style.css повинен розміщуватися в кореневій папці теми, а не в цій теці;
- /js/ – файли JavaScript;
- /images/ – зберігає вбудовані в тему зображення;
- /languages/ – каталог зберігає файли перекладів теми.

Є тільки два основних файли, які необхідні для активації теми:

- index.php – файл, який створює блог;
- style.css – стильове оформлення блогу.

Тим не менше, більшість тем включають в себе й інші файли. Ррозберемо основні з них:

- header.php – глобальний файл, який відображає мета-дані сторінки в розділі <head> та меню верхнього рівня;
- sidebar.php – цей файл відповідає за генерацію бічної колонки сайту. В основному тут виводяться: список рубрик (категорій), останні записи, теги, різні банери;
- footer.php – цей файл відповідає за виведення футера, нижнього меню, копірайту і закриває HTML-теги;

- index.php – це шаблон головної сторінки сайту. За замовчуванням цей файл відображає останні публікації та підвантажує інформацію з sidebar.php (сайдбару);
- single.php – відповідає за відображення окремих постів. Файл містить цикл, який викликає лише одну публікацію та формує її вивід;
- page.php – формує окремі (статичні) сторінки (наприклад, «Контакти», «Про нас» і т.п.);
- archive.php – цей файл відповідає за виведення сторінки архіву записів;
- category.php – формує шаблон сторінки, яка виводить публікації по категоріям;
- tag.php – шаблон сторінки, яка виводить список публікацій за тегами;
- comments.php – цей файл керує відображенням коментарів;
- functions.php – дозволяє додавати користувальницький код PHP та може впливати на основні елементи теми. Він додає функції та розширює можливості вашого сайту. Працює як плагін WordPress;
- style.css – основний файл CSS-стилів теми.

Це основні файли, які можуть бути присутніми в темі (шаблоні). Деякі теми можуть містити й інші файли, а деякі лиш пару-трійку. Все залежить від розробника теми, та бажання більш тонкого налаштування сайту. Існує великий вибір безкоштовних тем для водпрессу на різні тематики.

Преміум теми

Преміум теми для Вордпресс – це платні теми які часто розробоюються великими компаніями та мають вожливість гнучко налаштувати зовнішній вигляд та функціонал веб-ресурсу.

Основні різниці між платними та безкоштовними темами:

- Ціна яка може починатись від пьяти долларів сша до трьох ста та вище.
- Наявність додаткових можливостей для налаштування зовнішнього вигляду сайту, таких як конструктори сторінок.
- Наявністю підтримки сторонніх додатків, наприклад вукомерс та гравіті формс
- Наявністю технічної підтримки з боку розробників теми.

Не зважаючи на перелічени переваги, у зв’язку з тим що преміум теми містять у собі дуже багато додаткового функціоналу, який у свою чергу намагається задовільнити потреби різних цільових аудиторій, данні теми мають принципові недоліки роблячи неможливим автоматизацію процесу розробки.

Розглянемо основні недоліки:

- Відсутність якісті вихідного коду;
- Дуже низька продуктивність завантаження веб-сторінок;
- Після наповнення веб-сайту, якість його дизайну дуже відрізняється від тої що буда на вітрині біржі де ця тема була купленна. Загалом це пов’язано з тим що контент та зображення теми на вітрині було заповненно досвітченими дизайнерами, а купленний шаблон в більності виадках наповнюється покупцем який не має змоги передбачити якість ресурсу після його наповнення вхідним контентом;
- Неможливість масшабування та дорозроблення таких тем. Наприклад, інтеграція сторонніх сервісів які не передбачені розробниками данної теми.

1.3. CMS Drupal

Розглянемо систему керуванням звісту Drupal, як альнернативний метод проектування веб-сайтів.

Drupal — популярна вільна модульна система керування вмістом (СКВ, англ. CMS) з відкритим вихідним кодом, написана на мові програмування PHP та розповсюджується за ліцензією GNU.

Drupal використовують як back end фреймворк для різних веб-сайтіввід особистих блогів до корпоративних та державних сайтів. Drupal також використовується у системах управління знаннями та для ділової співпраці.

Drupal може працювати у таких популярних системах як Windows, Mac OS X, Linux, власне, на будь-якій платформі, яка підтримує роботу веб-сервера Apache, Nginx, Lighttpd або Microsoft IIS; також потрібна наявність системи керування базами даних MySQL/MariaDB, PostgreSQL 8.3, SQLite чи інші комерційні. Повні системні вимоги Drupal наведені на офіційному сайті.

До базового пакету системи, окрім модулів створення статичних сторінок (сторінок з постійною адресою) та нових статей входять модулі для організації блогів (електронних журналів користувачів), форумів (місць для інтернет-дискусій), «книг» (інформаційних добірок, праця над якими ведеться колективно), синдикації (імпорту новин з інших сайтів), модуль керування інформаційними блоками на сторінках, що полегшують керування їх виглядом, модуль керування меню.

Drupal підтримує різні теми оформлення та дозволяє створювати свої теми оформлення.

Спільнотою розробників Drupal'у створено багато додаткових модулів, серед яких варто згадати:

- модулі інтернаціоналізації (створення багатомовних сайтів);
- модулі керування файлами, що дозволяють викладати на сайтах звукові та відео-файли;
- модулі категоризації вмісту, модулі організації користувачів у групи та спільноти.

Основні можливості

У дистрибутив системи входить набір модулів, що дають наступні можливості:

- збір інформаційних стрічок (RSS, RDF, Atom);
- ведення блогів, підшивань і форумів;
- створення форм для відправки повідомлень;
- локалізація системи;
- перейменування посилань (призначення посиланням зрозумілих і зручних псевдонімів);
- проведення опитувань;
- призначені для користувача профілі, що налаштовуються;
- пошук за змістом (за зміст вважається і повідомлення на форумах, і сторінки, і будь-які інші призначені елементи);
- ведення журналу статистики (відвідуваності);
- таксономія (впорядковування матеріалу за категоріями);
- формування сторінок з матеріалами в різних формах і форматах подання та інші.

Механізми рубрикації

Кожен документ сайту може входити в одну або кілька рубрик. Самі ж рубрики можуть складати списки або складні ієрархічні структури довільної вкладеності (з множинними предками і перехресними посиланнями елементів).

Інтеграція всіх компонентів

Можлива наскрізна рубрикація за всіма типами документів сайту (наприклад список ключових слів, загальний для форумів та блогів). Форум із виводом цікавих новин на головну сторінку або сайт новин із блогами та відеопрезентації — все це можна укласти в єдиний рубрикатор (або декілька рубрикаторів) і це буде виглядати частинами єдиного сайту, а не розрізненими сторінками об'єднаними лише загальним дизайном.

Готові рішення типових завдань

Сайт новин, сайт-візитка компанії, блог або форум — такі сайти можна побудувати, користуючись тільки модулями рушія, що йдуть у поставці, потрібно тільки включити відповідні модулі, налаштувати їх і перенести сайт на хостинг.

Навігація і пошук

Для зручності доступу до архівних матеріалів служать рубрикація контенту і пошук з урахуванням видів контенту, рубрик та вмісту. Документи зберігають незмінні посилання весь час свого життя (т. зв. перманентні посилання). Також за допомогою коротких посилань і псевдонімів сайт набуває запам'ятовуються імена розділів і окремих сторінок, які не використовують спеціальних символів і тому добре індексовані пошуковими системами.

Таксономія

Це оригінальна методика притаманна Drupal для завдання структури сайту, спосіб відділити структуру від подання. За допомогою таксономії

можна визначити довільну кількість рубрик, в яких будуть надалі міститися матеріали сайту.

Ці рубрики можуть бути представлені як плоскі списки або ієрархічні структури довільної вкладеності (як деревоподібні, коли елемент має тільки одного з батьків в ієрархії, так і довільні, коли елемент може мати відразу декількох батьків).

Інша парадигма з'явилася зі створенням в Drupal модуля Content Construction Kit (CCK). (З виходом версії Drupal 7 — перенесено в ядро.) CCK дозволяв доповнювати документи новими полями різних типів — від полів вводу URL і email, до полів зберігання і відображення мультимедійних файлів. Також за допомогою додаткових модулів до CCK (наприклад Node reference) можна організувати зв'язок між документами, не використовуючи механізм таксономії.

В Drupal 7 майже весь функціонал CCK перенесений в ядро системи. В модулі CCK лишилися хелпери (наприклад підтримка PHP коду).

На Рис. 4 представленна реляційна схема таблиц Таксономіі у базі данних спроектованної на базі Drupal.

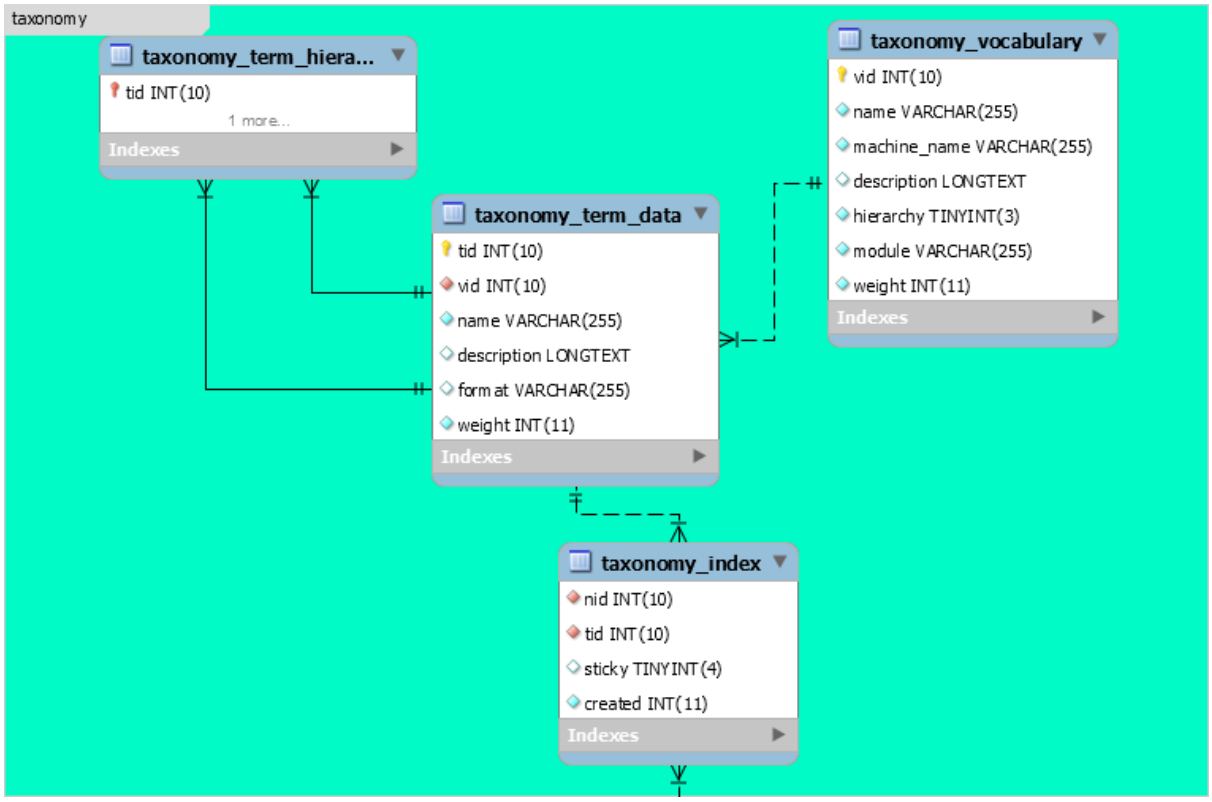


Рис. 4

Механізми розширення функціоналу

Drupal має модульну архітектуру з компактним ядром, що надає API, до якого можуть звертатися модулі. Стандартний набір модулів включає такі функції, як новинна стрічка, блог, форум, завантаження файлів, збирач новин, голосування, пошук тощо. Дизайн сайту змінюється також за допомогою спеціальних модулів — «тем оформлення».

Модулі

Кілька прикладів модулів, які можна завантажити в репозиторії на drupal.org:

- Ad — система для управління показами рекламних банерів;

- Captcha — захисний механізм картинок «CAPTCHA», що використовується при реєстрації;
- Ecommerce, Ubercart, Drupal Commerce — системи електронної комерції;
- FCKeditor, CKEditor, Tinymce — візуальні редактори;
- Gallery — інтеграція з галереєю зображень Gallery2;
- LDAP integration — інтеграція з LDAP;
- mailhandler, listhandler — можливість публікації матеріалів сайту через поштовий інтерфейс і з поштових розсилок;
- Project — ведення проектів, включає багтрекер і інтеграцію з CVS і Subversion;
- SPAM — блокування спаму;
- Swish, Sphinxsearch — інтеграція з рушіями пошуку Swish і Sphinx;
- View — візуальний побудовник запитів до матеріалів і відображення їх в різних ракурсах;
- WebForm — гнучкий модуль для швидкого проектування інтерактивних форм (опитування, зворотний зв'язок, голосування, та оформлення замовлень).

Теми оформлення

Дизайн сайту на Drupal будується на основі змінних тем оформлення. Немає єдиної схеми побудови дизайну. Натомість Drupal дає можливість використовувати різні «рушії тем», що використовують шаблони, зручні для редагування (шаблони XML у рушіях xtemplate і Smarty або шаблони на HTML і вбудований PHP у рушії phptemplate тощо), або створювати теми оформлення безпосередньо звертаються до API Drupal. У комплект

поставки Drupal включений рушій тем на основі phptemplate і кілька прикладів тем. Інші пакети тем можна завантажити на сайті проекту.

Починаючи з версії 4.7 Drupal підтримує технологію Ajax для динамічного підвантаження вмісту без повного оновлення сторінок. У версії 5 для роботи з JavaScript додана бібліотека JQuery. З версії 6.0 з'явилося створення форм за допомогою AHAH (без повного перезавантаження сторінок сайту).

Структура вхідного коду

Розглянемо анатомію двигуна Drupal і за що відповідають ті, чи інші файли та папки.

- /core - файли, необхідні для використання поза ядрами Drupal (ядро), за винятком файлів, які мають явну причину включення в базовий (/) каталог;
- /libraries - зовнішні бібліотеки сторонніх організацій, якими користується Drupal, такі як редактор WYSIWYG. Ця папка не включена в ядро, але використовується з багатьма наданими модулями;
- /modules - каталог, в який переходять усі власні (створені нами) та внесені (створені громадою) модулі. Розбиття цього розділу на додаток до підкаталогів та на замовлення може полегшити відстеження модулів. Ви можете створити підпапки для організації, щоб відповідати стандартам вашої розробки, зберігання, використання;
- ./profile - профілі встановлення та спеціальні установки;

- sites/[domain OR default]/{modules,themes} - модулі та теми, визначені для сайту, можна перемістити в ці каталоги, щоб уникнути їх відображення на кожному сайті. Ідентично для Drupal 7;
- sites/[domain OR default]/files - зберігання файлів для конкретних сайтів. Сюди входять файли, завантажені користувачами (наприклад, зображення) та конфігурація сайту (активна та поетапна) ;
- /themes - усі внесені та власні теми та підтеми. Зауважте, що підтеми вимагають також встановити базову тему;
- /vendor - зовнішні бібліотеки, від яких залежить ядро Drupal (приклади Symfony, Twig).

Також розглянемо структуру папки /core

- /core/assets - зовнішні бібліотеки, що використовуються ядром (включає jQuery, підкреслення, модернізатор тощо ;
- /core/includes - функціонал базового рівня, який Drupal використовує через інші / основні папки;
- /core/lib - Drupal основні класи;
- /core/misc - код Frontend, від якого залежить ядро Drupal;
- /core/modules - Drupal модулі ядра;
- /core/profiles - профілі установки Drupal. Це мінімальні, стандартні, тестувальні та тестуючі багатомовні профілі установки;
- /core/scripts - сценарії інтерфейсу командного рядка (CLI), які в основному використовуються розробниками;
- /core/tests - Тести на ядро Drupal;

- /core/themes - Основні теми Drupal.

Недоліки системи Drupal

У системі Drupal слабке використання об'єктних можливостей PHP. API Drupal практично не використовує наявні в PHP можливості ООП. Розробники аргументують це слабкою реалізацією ООП у мові (особливо до версії PHP 5). Об'єктна модель в Drupal присутня, але в дещо нетрадиційному для PHP вигляді. Тим часом, для оптимізації розробки веб-сайтів нам потрібно використовувати можливости ООП у мові программування PHP.

До недоліків Drupal можна віднести відсутність зворотньої сумісності API при досить високій динаміці розробки проекту.
Практично в кожному релізі відбуваються зміни API, коли поряд з додаванням нових функцій прибираються деякі старі або змінюються параметри виклику функцій. Це призводить до необхідності розробникам сторонніх модулів адаптувати їх для роботи з новими версіями Drupal.

Проте зміни API і процедура адаптації модулів до нових версій описуються в документації до кожного релізу, також завжди пропонується механізм автоматизованого апгрейда ядра системи на нову версію.

Плюс даної схеми розробки — немає необхідності тягти з версії у версію програмний шар сумісності зі старими API, що полегшує поточний код системи.

З іншого боку, ставлячи на меті автоматизувати процес проектування та розробки веб-сайтів, відсутність зворотньої сумісності API робить данну

систему не надійною, тому що етапи проектування можуть бути змінені у будь-який час, що потребує додаткових ресурсів на ознайомлення.

Висновки до розділу 1

У цому розділі ми оглянули методи розробки вебсайтів на базі Wordpress та Drupal. Ці системи надають можливості швидкого проектуваняя веб сайтів тому що включають у себе досить багато функціоналу.

Порівнюючи архітектуру та можливості вище згаданних систем, можна зробити висновок що система Wordpress більше підходить для автоматизації розробки веб-сайтів, але незважаючи на те що існує безліч готових шаблонів тем для Wordpress, загалом є необхідніть у розробці персональної теми та плагінив, зовнішній вигляд та функціонал, яких будет задовільняти бізнес логіці проектів.

У наступному розділі ми розглянемо усі необхідні інструменти та технології для розробки кастомної теми призначеної для створення веб-сайтів на туристичну тематику.

РОЗДІЛ 2: ЗАСОБИ ПРОЕКТУВАННЯ ТА АВТОМАТИЗАЦІЇ РОЗРОБКИ ВЕБ-САЙТІВ

1.1. Типи систем планування проектів

Зазвичай процесс розробки веб-сайту включає у себе деякі етапи, тож перед тим як автоматизувати цей процесс, розглянемо типи та засоби планування безпосередньо самого процессу розробки.

Типи планування проектів

Waterfall є одним із найстаріших методів, але все ще використовується багатьма командами розробників. Цей стиль передбачає роботу у хвилях, при цьому кожен крок сильно залежить від того, що передує.

Час, витрачений напочатку виробничого циклу програмного забезпечення, може зменшити витрати на більш пізніх етапах. Наприклад, проблему,

виявлену на ранніх стадіях (наприклад, специфікація вимог), виправити дешевше, ніж та сама помилка, яка виявилася пізніше в процесі (з коефіцієнтом від 50 до 200).

У звичайній практиці методології водоспаду призводять до розкладу проектів з 20–40% часу, вкладеного на перші дві фази, 30–40% часу на кодування, а решта присвячена тестуванню та впровадженню. Фактична організація проекту повинна бути високоструктурованою.

Більшість середніх та великих проектів включатимуть детальний набір процедур та контролю, які регулюють кожен процес проекту.

Наступним аргументом для моделі водоспаду є те, що він робить акцент на документації (такі як вимоги та проектні документи), а також на вихідний код.

У менш ретельно розроблених та задокументованих методологіях знання втрачаються, якщо члени команди виїжджають до завершення проекту, і проекту може бути важко відновити втрати.

Якщо присутній повністю працюючий проектний документ (як це має намір Великий фронт проектування та модель водоспаду), нові члени команди або навіть зовсім нові команди повинні мати можливість ознайомитись, прочитавши документи. На Рис. 5 схематичне відображення методу проектування Waterfall.

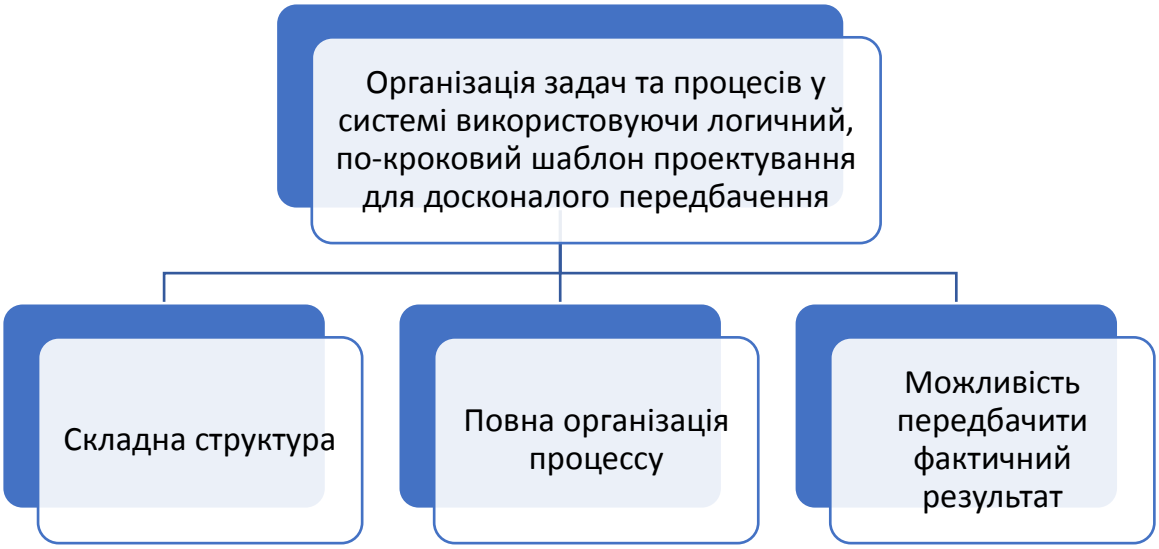


Рис. 5

Модель водоспаду забезпечує структурований підхід; сама модель прогресує лінійно через дискретні, легко зрозумілі та пояснювані фази і тому легко зрозуміти; він також забезпечує легко визначити основні етапи в процесі розробки. Можливо, з цієї причини модель водоспаду використовується як початковий приклад моделі розробки в багатьох текстах та курсах програмного забезпечення.

Agile - це ітеративний підхід до управління проектами та розробки програмного забезпечення, який допомагає командам швидше та з меншими головними болями доставляти цінність своїм клієнтам. Замість того, щоб робити все на старт "великого удару", спритна команда забезпечує роботу невеликими, але витратними кроками.

Вимоги, плани та результати оцінюються постійно, тому команди мають природний механізм швидкого реагування на зміни. На Рис. 6 схематичне відображення методу проектування Agile.

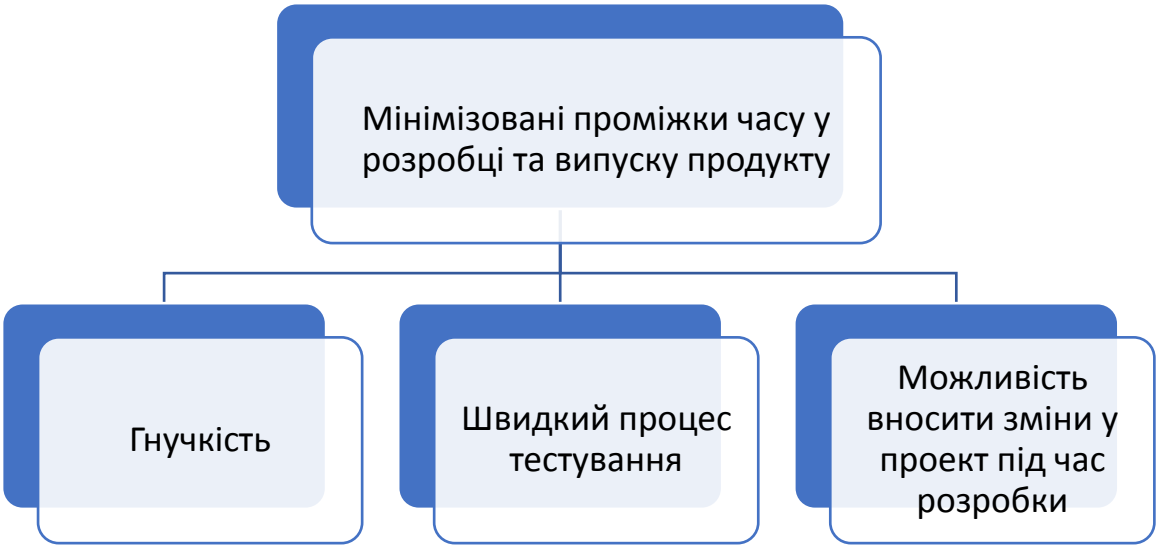


Рис. 6

Scrum – це під-тип **Agile**, де характерною рисою є використання спринтів для виконання проектів невеликими частинами, часто грунтуючись на місячному графіку.

Канбан - ще один варіант спритного управління проектами. На відміну від Scrum, який орієнтований на часові п'єси, Kanban все стосується організації. Для цього Канбан переглядає насамперед кількість завдань, які входять у будь-який процес, і на те, як їх можна впорядкувати, зменшити тощо.

Метод **PRINCE2** часто використовується приватними секторами в уряді і орієнтований на ефективність та мінімізацію ризиків та помилок. Цей метод, орієнтований на деталі, стосується об'єднання проектів на етапи, що базуються на продуктах, які можна вирішувати один за одним, гарантуючи, що жоден камінь не буде перевернутий ніде в процесі.
На Рис. 7 схематичне відображення методу проектування Prince2.

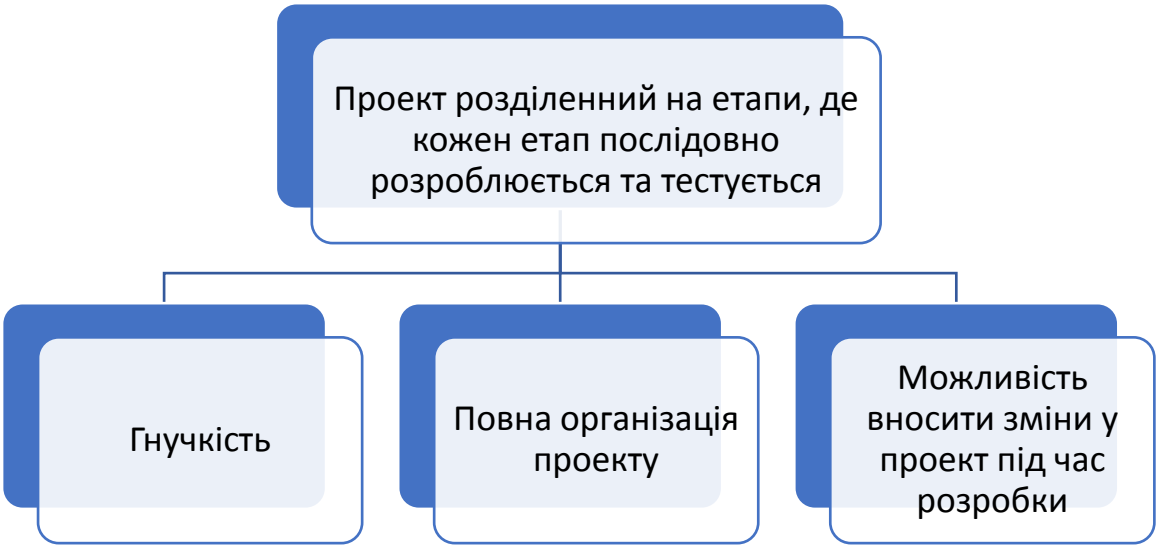


Рис. 7

Засоби для планування проектів

Загалом розглянемо декілька популярних платформ для керування проектами.

Jira - продукт відстеження релізів, який дозволяє відслідковувати помилки та спритне управління проектами.

Основні фунції Jira

- Підтримка Канбан;
- Підтримка Scrum;
- Можливість переглядати процес розробки проекту вудображеному у проміжках часу;
- Можливість строворювати звіти.

На Рис. 8 відображен внутрішній вигляд системи Jira.

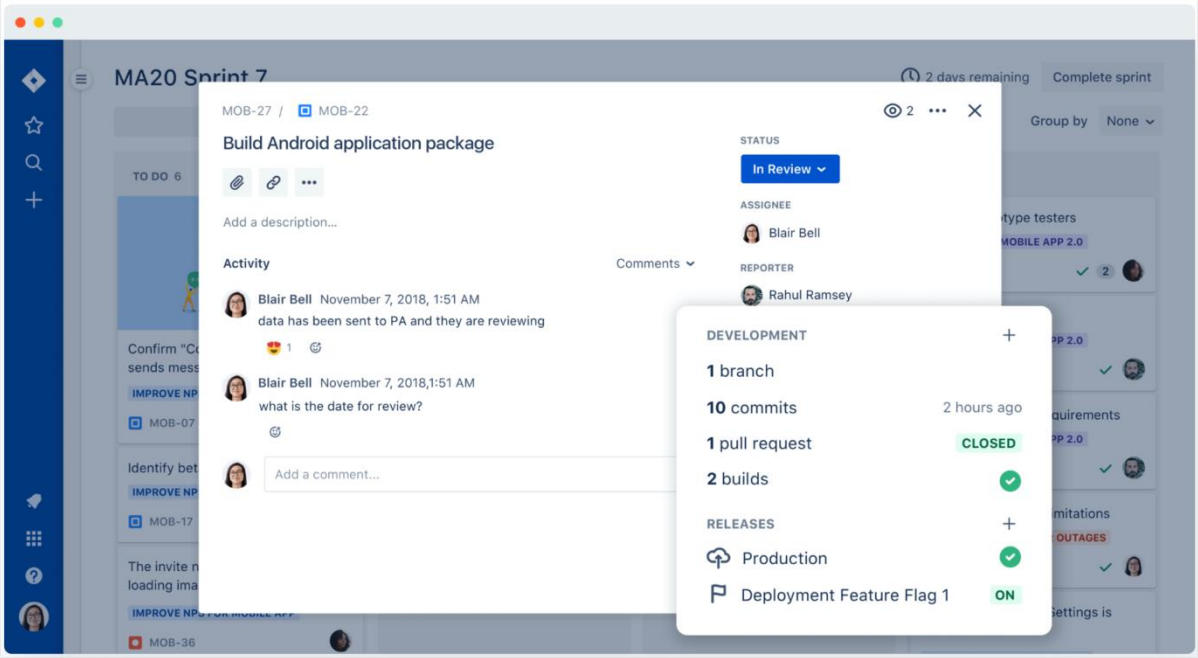


Рис. 8

Trello - це веб-додаток для створення списків в стилі Канбан. Користувачі можуть створювати свої дошки завдань з кількох стовпців і переміщувати завдання між ними.

Зазвичай стовпці містять статуси завдань: «Робити», «Проробляється», «Готово». На Рис. 9 відображен внутрішній вигляд дошки Trello.

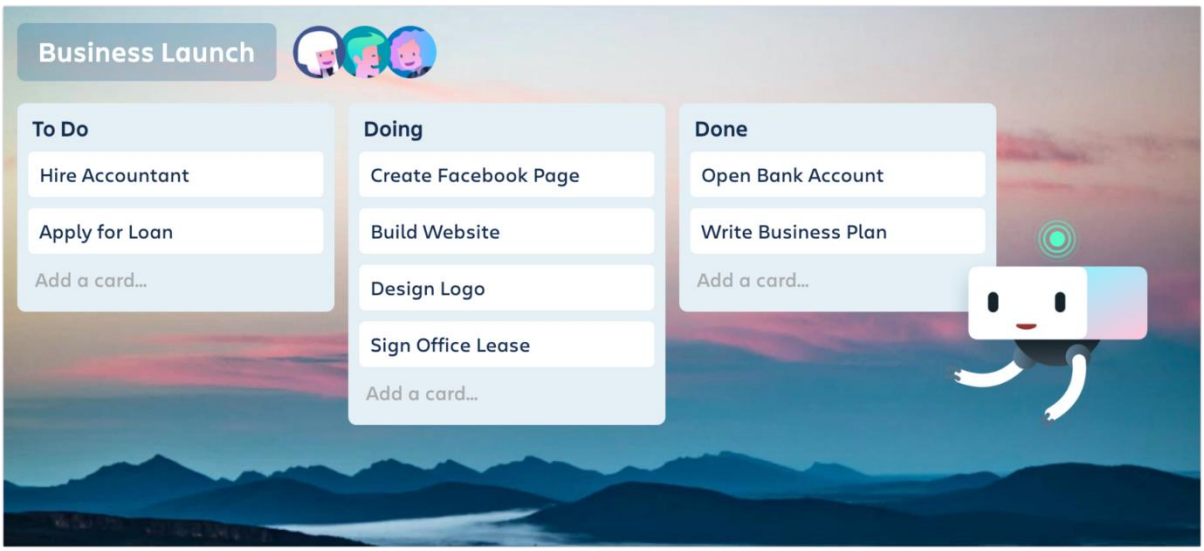


Рис. 9

Інструмент може бути для особистого та ділового використання. Trello має різноманітні робочі та особисті потреби, включаючи управління нерухомістю, управління проектами програмного забезпечення, дошки оголошень шкіл, планування уроків, бухгалтерський облік, веб-дизайн, ігри та управління справами в адвокатському бюро.

Багатий API, а також можливість входу в електронну пошту дозволяє інтегруватись з корпоративними системами або з хмарними послугами інтеграції.

Основні фунції Trello

- Підтримка Канбан;
- Можливість ствоювати команди та сортувати дошки між ними;
- Можливість автоматизувати процеси розробки за допомогою модулю Butler;
- Можливість додавати кліентів на дошки без необхідності мати людини у команді.

На Рис. 10 відображен внутрішній вигляд задачі з дошки Trello.

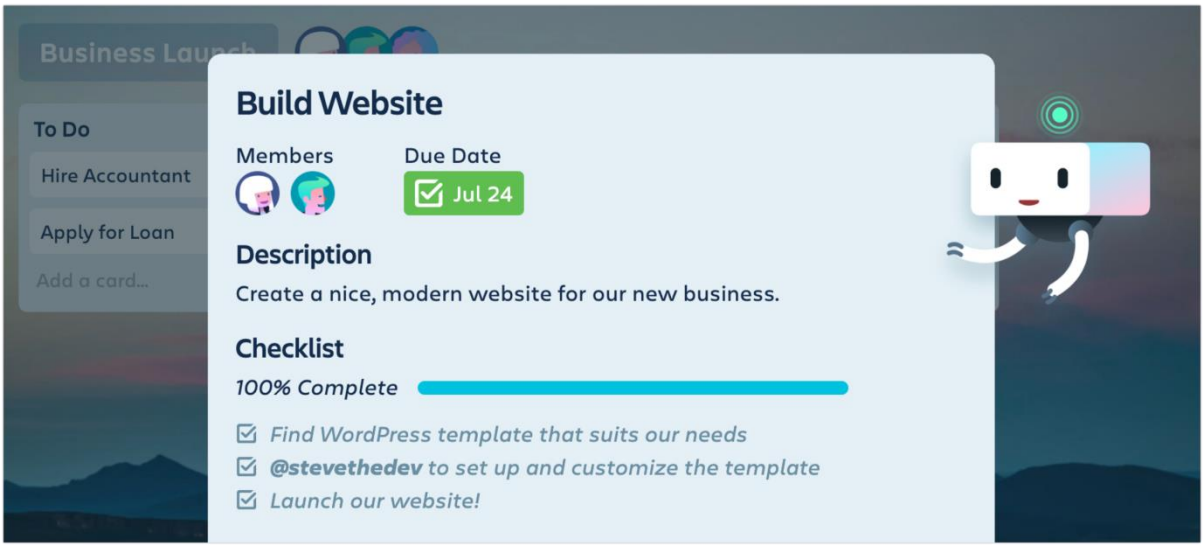


Рис. 10

Ключова можливість модулю Butler для автоматизації керування розробкою

Можливість додати довільну кнопку до внутрішнюго інтерфейсу задач, наприклад «Відправити на тестування», яка перенесе задачу у стовбець «Тестування» та повідомить необхідних людей про це, тож розробник може відзвітувати про досягнення одним кліком.

Asana - це веб- та мобільний додаток, призначений допомогти командам в організації, відстеженні та керуванні їх роботою. Відкритий API Asana надає засоби для програмного зчитування інформації в Asana, введення інформації в Asana та створення автоматизації в Asana. Загальні випадки використання включають автоматизацію повторюваних завдань, ланцюжок процесів, автоматизацію звітування про завдання та проекти та синхронізацію з базами даних чи іншими інструментами.

API Asana - це інтерфейс RESTful, що дозволяє оновлювати та отримувати доступ до більшості даних на платформі. Він надає передбачувані URL-

адреси для доступу до ресурсів і використовує вбудовані функції HTTP для отримання команд та повернення відповідей. Це дозволяє легко спілкуватися з найрізноманітнішими середовищами: від утиліт командного рядка до плагінів браузера до нативних програм.

На Рис. 11 зображено внутрішній вигляд списку задач у платформі Asana.

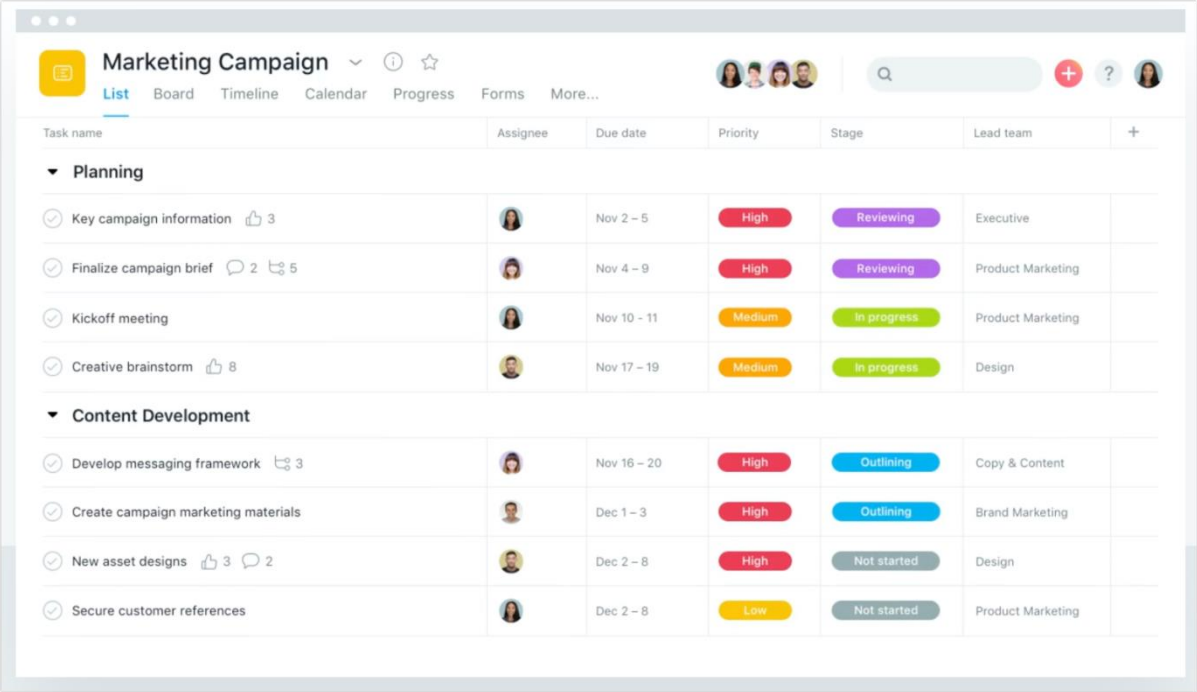


Рис. 11

1.2. Налаштування локального хосту

При розробці власного шаблону теми, є необхідність у налаштування робочої середи яка повністю відповідала би середі на справжньому продакшені. Для цього зручніше за все використовувати локальний веб-сервер на своєму комьютері. Характерні переваги цього способу є відсутність обмежень на розмір сайту, використання процессорного часу та оперативної памьяті серверу, а головне – є можливість відпрацьовувати програмний код безпосередньо на своєму комьютері без необхідності зозвертати проект на сервері.

Данний шаблон теми розроблюється на операційній системі Mac OS X яка використовує сервер Apache, увімкнув який, ми маємо змогу працьовати над статичними сайтами, але для розробки динамічних сайтів, нам необхідно вміти задавати дуже багато конфігурацій для вбудованого серверу MySQL.

Для того щоби з економити час, та розвертати локальний веб-сервер з усіма необхідними налаштунками ми використовуємо спеціальний налаштунок під назвою MAMP. Абревіатура MAMP розшифровується як Macintosh, Apache, MySQL и PHP.

Існує дві версії застосунку: безкоштовна MAMP та платна MAMP PRO, які відрізняються одна від одної функціональністю та принципом доступу до управління сервером. У стандартний дестрибутив уходіть обидві версії.

Везкоштовна версія MAMP меньш гнучка ніж розширена. Управління налаштунками робиться через вікно програми, а робота із веб-сервером через веб-інтерфейс. Також у безкоштовній версії є можливість установити порти для Apache та MySQL, обрати версію PHP, оптимізатор (XCache, APC, eAccelerator) та налаштувати Root-директорію для проекту.

Навідміну від інших популярних шляхів для роботи із локальним веб-сервером, обидві версії MAMP та MAMAP PRO вміють при своєму запуску також запускати веб-сервер, а при виході – зупиняти. На рис. 12 зображений інтерфейс программи.

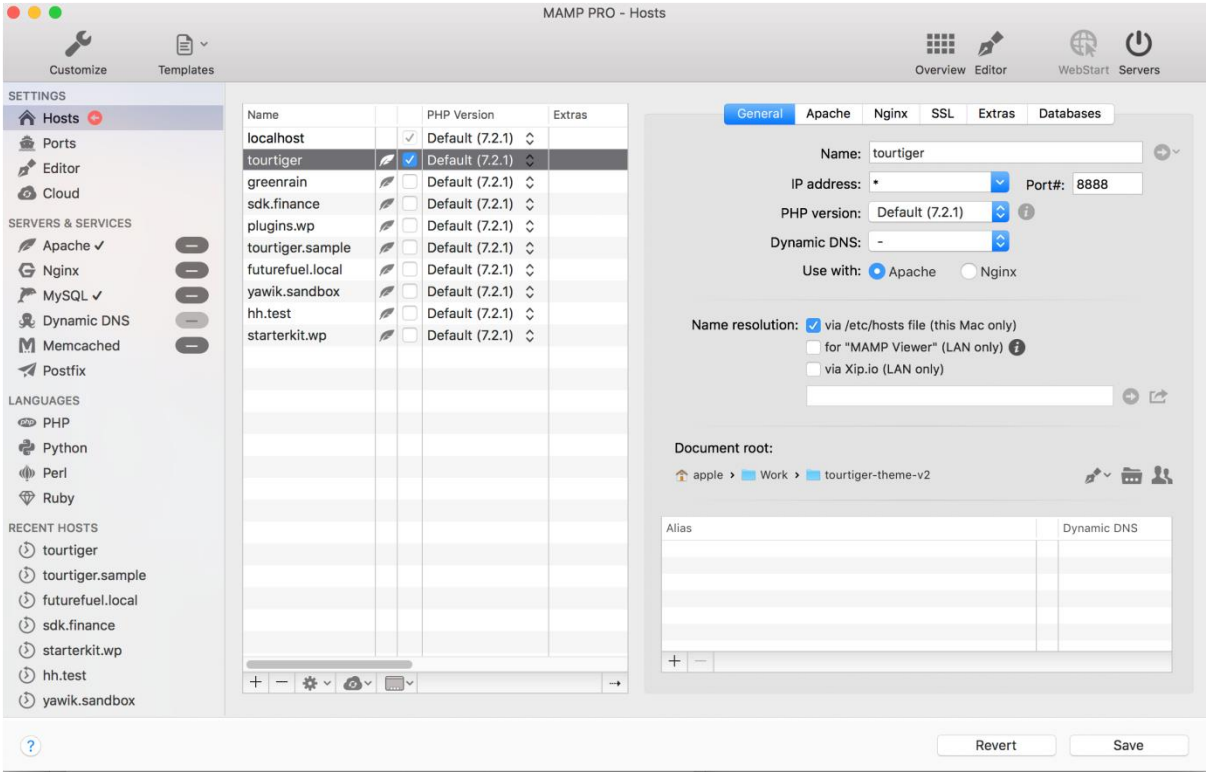


Рис. 12

У MAMP PRO налаштування та управління веб-сервером робиться через графічне вікно конфігуратору, опцій у якому на багато більше. У верхній панелі вікна розташовані кнопки для ручного запуску та зупинки роботи веб-серверу, екрат стану Apache, MySQL, а також клієнту DynDNS.

Комплексно порівнявши безкоштовну та розширену версії застосунку для роботи із локальним веб-сервером MAMP, було прийнято рішення використовувати розширену версію застосунку для розробки шаблону теми для вордпресс. Нижче розглянемо конфігурації MAMAP PRO.
Сервер та основні налаштування.

На цій вкладці ми налаштовуємо порти для Apache та MySQL, є можливість використовувати свої порти або ті що задаються за заумовчанням.

Apache

- У данній вкладці є можливість включати та відключати окремі модулі Apache. Ми використовуємо це, коли є необхідніть розробляти під специфічні веб-сервери, які не підтримують ті або інакші модулі.
- Також є можливіть задати папку куди будуть складатися звіти о помилках.

MySQL

- Задання/зміна головного паролю MySQL.
- Налаштування папки куди будуть зберігатись звіти з помилками.
- Доступ до phpMyAdmin.

PHP

- Вибір версії PHP та конфігурація Zend Optimizer.
- Для PHP додатково можна обрати рівень попереджень та спосіб їх відображення (на єкран або у лог-файл). Ми обираему перший варіант.

Основні налаштування хостингу

- У MAMP PRO можна создавати скільки завгодно віртуальних хостив, це дуже зрочно для тестингу шаблону теми у різних середовищах.
- Також для кожного віртуального хостингу ми використовуємо різні папки в операційній системі що надає нам більше гнучкості.

1.3. Інтелектуальний редактор коду PhpStorm

PhpStorm – це інтегрована середа розробки на PHP з інтелектуальним редактором, який розуміє код, аналізує його та підтримує PHP 7.2-5.3 для

написання класичних та сучасних проектів, забезпечує розумне автодоповнення коду, рефакторинг, запобігання помилок та підтримує змішування мов програмування.

Сотні інспекцій піклуються про верифікації коду, аналізуючи проект цілком під час розробки.

Підтримка PHPDoc, code (re) arranger, форматування коду з конфігурацією стилю коду і інші можливості допомагають розробникам писати охайний і легко-підтримуваний шаблон теми для вордпрессу.

Підтримуються передові технології веб-розробки, включаючи

- HTML5
- CSS
- Sass
- SCSS
- Less
- Stylus
- Compass
- CoffeeScript
- TypeScript
- ECMAScript Harmony
- шаблони Jade
- Zen Coding
- Emmet
- JavaScript.

Інтелектуальний редактор PHP коду з підсвічуванням синтаксису, автодоповнення коду, розширеними налаштуваннями форматування коду, запобіганням помилок нальоту.

На рис. 13 зображений приклад повідомлення про помилку у написанні коду.

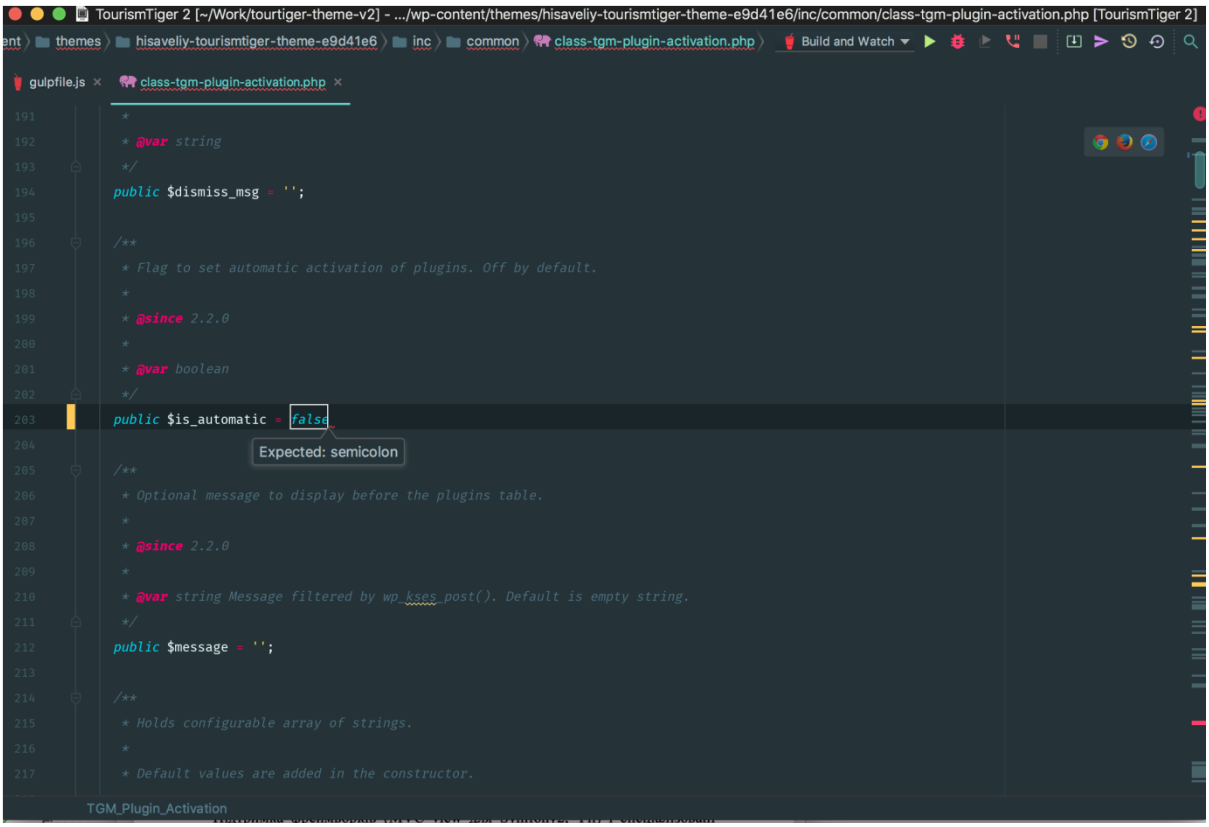


Рис. 13

Підтримує PHP 7.2-5.3, генератори, співпрограми і усі синтаксичні поліпшення. PHP рефакторингом, code (re) arranger, детектор дубльованого коду.

Підтримка Vagrant, Composer, вбудований REST клієнт, Command Line Tools, SSH консоль

Підтримка фреймворків (MVC view для Symfony2, Yii) і спеціалізовані плагіни для провідних PHP фреймворків:

- Symfony
- Magento
- Drupal
- Yii
- CakePHP
- WordPress
- Joomla

Підтримка стилів коду, вбудовані стилі PSR1 / PSR2, Symfony2, Zend, Drupal та інші.

Інтеграція з системами управління версіями, включаючи уніфікований інтерфейс.

Віддалене розгортання додатків і автоматична синхронізація з використанням FTP, SFTP, FTPS і ін.

Live Edit: зміни в коді можна миттєво переглянути в браузері без перезавантаження сторінки.

Інструменти роботи з базами даних, SQL редактор. Крос-платформенность (Windows, Mac OS X, Linux)

1.4. Хмарні сховища коду та його версіювання

Ми використовуемо Git для версіювання нашого коду. Це дозволяє нам випускати шаблон теми частинами, поступово наповнюючи його новим функціоналом.

Головна відмінність Git'у від будь-яких інших контроллерів версій (наприклад, Subversion і їй подібних) - це те, як Git дивиться на свої дані. В принципі, більшість інших систем зберігає інформацію як список змін (патчів) для файлів. Ці системи (CVS, Subversion, Perforce, Bazaar і інші) відносяться до збережених даних як до набору файлів і змін, зроблених для кожного з цих файлів в часі, як показано на рис. 14.

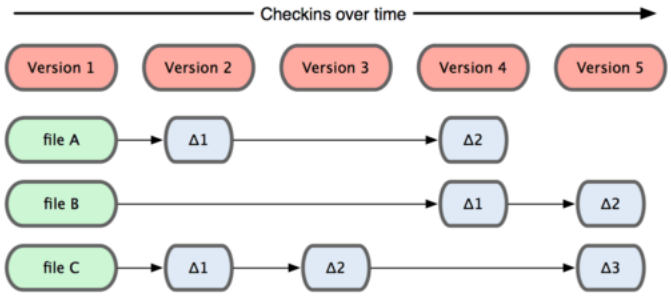


Рис. 14

Git не зберігає свої дані в такому вигляді. Замість цього Git вважає збережені дані набором зліпків невеликий файлової системи. Кожен раз, коли ми фіксуємо поточну версію теми для вордпресс, Git, по суті, зберігає зліпок того, як виглядають всі файли проекту на поточний момент.

Заради ефективності, якщо файл не змінювався, Git не зберігається файл знову, а робить посилання на раніше збережений файл. Те, як Git підходить до зберігання даних.

Це важлива відмінність Git'у від практично всіх інших систем контролю версій. Через нього Git змушений переглянути практично всі аспекти контролю версій, які інші системи перейняли від своїх попередниць. Git більше схожий на невелику файлову систему з неймовірно потужними інструментами, що працюють поверх неї, ніж на просто ВКВ.

Три стани

У Git'і файли можуть перебувати в одному з трьох станів: зафіксованому, зміненому і підготовленому.

- Зафіксований означає, що файл вже збережено у нашій локальній базі.
- До змінених відносяться файли, які змінилися, але ще не були зафіксовані.
- Підготовлені файли - це змінені файли, відмічені для включення в наступний Ком.

Таким чином, у нашому проекті, є три частини: каталог Git'у (Git directory), робочий каталог (working directory) і область підготовлених файлів (staging area).

Каталог Git'у - це місце, де Git зберігає метадані та базу даних об'єктів нашого проекту. Це найбільш важлива частина, і саме вона копіюється, коли ми клонуєте репозиторій з темою з іншого комп'ютера або репозиторію.

Робочий каталог - це витягнута з бази копія певної версії проекту. Ці файли дістаються з стислій бази даних в каталозі і поміщаються на диск для того, щоб ми мали змогу їх переглядати і редагувати.

Область підготовлених файлів - це звичайний файл, зазвичай зберігається в каталозі, який містить інформацію про те, що повинно увійти в наступну версію шаблону. Іноді його називають індексом (index), але останнім часом стає стандартом називати його областю підготовлених файлів (staging area).

Наш робочий процес з використанням Git'у виглядає приблизно так:

1. Ми вносимо зміни в файли в своєму робочому каталозі.
2. Готуємо файли, додаючи їх зліпки в область підготовлених файлів.
3. Робимо Комміт, який бере підготовлені файли з індексу і поміщає їх в каталог на постійне зберігання.

Якщо робоча версія файлу збігається з версією в каталозі Git'у, файл вважається зафіксованим. Якщо файл змінений, але доданий в область підготовлених даних, він підготовлений. Якщо ж файл змінився після вивантаження з БД, але не був підготовлений, то він вважається зміненим. Більш детальна інформація представлена на рис. 15

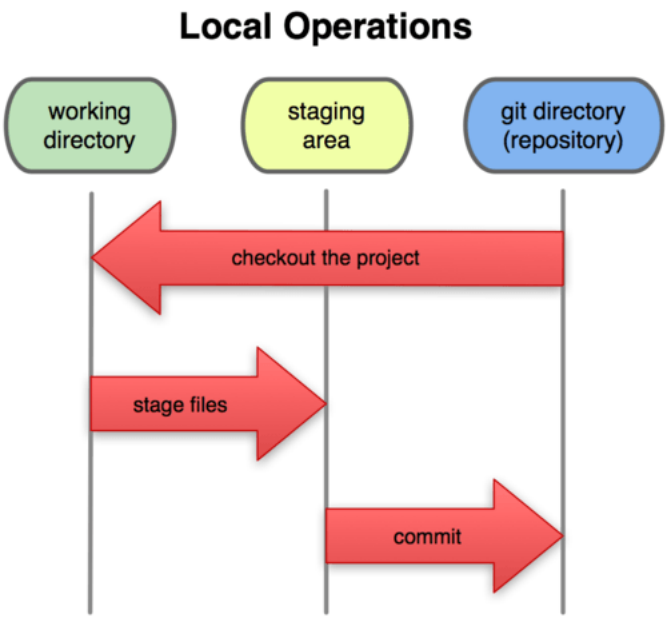


Рис. 15

1.5. Технології та стандарти роботи зі статичними файлами

HTML5 — наступна версія мови HTML. До складу робочої групи з HTML5
увійшли AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera та кілька сотень інших виробників.

Специфікації HTML5 не обмежуються тільки розміткою і включають в себе низку веб-технологій, котрі у сукупності формують відкриту Веб-платформу — програмне оточення для роботи крос-платформових застосунків, здатних взаємодіяти з обладнанням, і які підтримують засоби для роботи з відео, графікою і анімацією, що надає розширені мережеві можливості.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3.

Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Для покращення архітектури каскадних таблиць стилів ми використовуємо пре-процессор LESS. На рис. 16 представлений CSS синтаксис та на рис. 17 – синтаксис LESS.

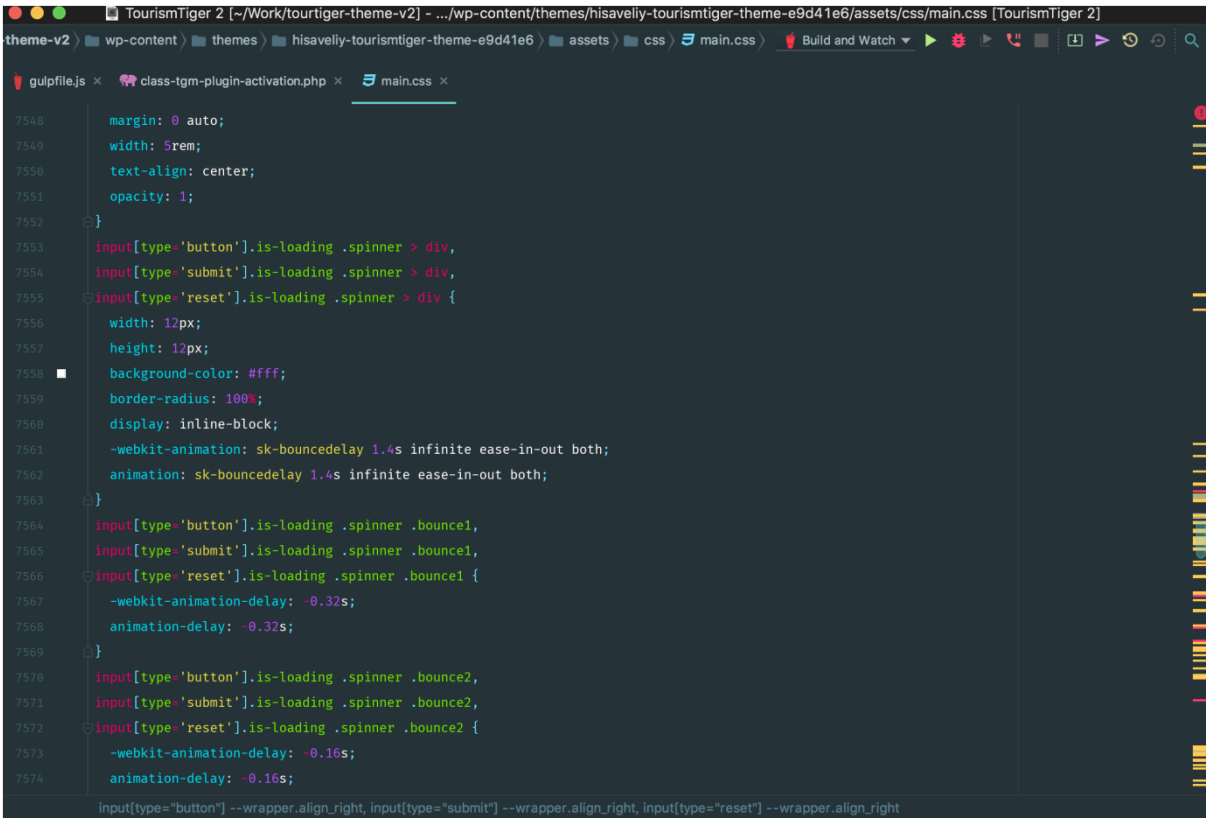


Рис. 16

LESS - це надбудова над CSS, що означає - будь-який CSS код - це валідний LESS, але додаткові елементи LESS не працюватимуть в

простому CSS. Це чудово, тому що існуючий CSS вже є працездатним LESS кодом, що зменшує поріг входження в нову технологію.

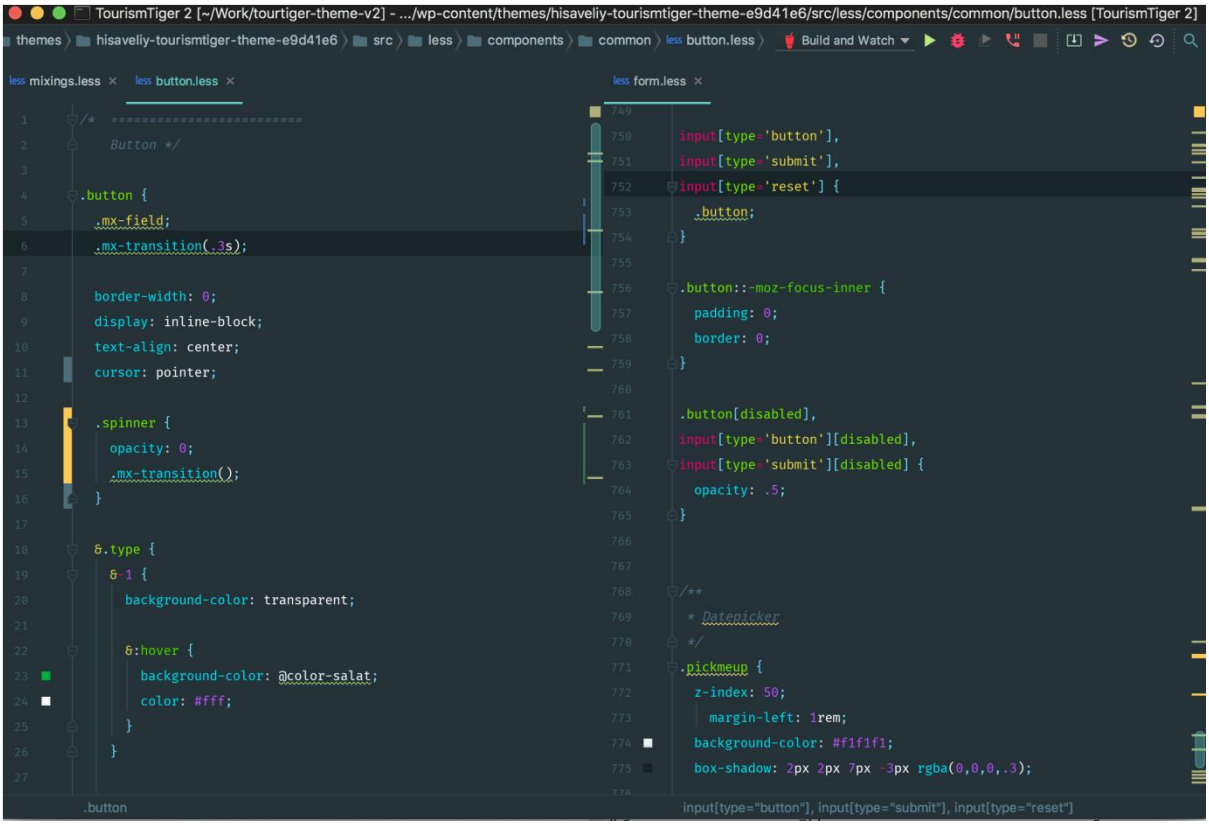


Рис. 17

LESS додає багато потрібних динамічних властивостей в CSS. Він вводить змінні, операції, function-like елементи і домішки. Можливість писати таблиці стилів модульно позбавить вас від багатьох клопоту.

Використання LESS у розробці теми для Вордпресс.

Існує два способи використання LESS:

- Створення LESS файлу і конвертування його за допомогою Javascript на льоту;

- Компілювання його заздалегідь і використання отриманного CSS файлу.

Ми використовуємо другий варіант. Для того щоб не конвертувати код при кожному завантаженні сторінки ми використовуємо результуючий CSS файл.

Змінні в LESS працюють так само як в PHP, JS і в більшості інших мов програмування. У розробці клієнтської частини найої теми ми використовуємо змінні значення, і потім використовуємо змінні замість самого значення щоразу, коли вам це потрібно. Наприклад, ми використовуємо змінні для:

- Кольорів
- Сімейств шрифтів
- Розмірів вікна браузера
- Відступів між елементами

А для надання веб-ресурсам розробленним за допомогою нашого шаблону теми додаткової інтерактивності ми використовуємо JavaScript ES6.

Основні поняття JavaScript та ECMAScript.

JavaScript створювався як скриптова мова для Netscape. Після чого він був відправлений в ECMA International для стандартизації (ECMA - це асоціація, діяльність якої присвячена стандартизації інформаційних і комунікаційних технологій). Це призвело до появи нового мовного стандарту, відомого як ECMAScript.

Подальші версії JavaScript вже були засновані на стандарті ECMAScript. Простіше кажучи, ECMAScript - стандарт, а JavaScript - найпопулярніша реалізація цього стандарту.

ES - це просто скорочення для ECMAScript. Кожне видання ECMAScript отримує абревіатуру ES з подальшим його номером. Всього існує 8 версій ECMAScript. ES1 була випущена в червні 1997 року, ES2 - в червні 1998 року, ES3 - в грудні 1999 року, а версія ES4 - так і не була прийнята. Не будемо заглиблюватися в ці версії, так як вони морально застаріли, а розглянемо тільки останні чотири.

ES5 був випущений в грудні 2009 року, через 10 років після виходу третього видання. Серед змін які ми використовуємо у шаблоні можна відзначити:

- підтримку суворого режиму (strict mode);
- аксессор getters і setters;
- можливість використовувати зарезервовані слова в якості ключів властивостей і ставити коми в кінці масиву;
- нову функціональність в стандартній бібліотеці;
- підтримку JSON.

Версія ES6 / ES2015 вийшла в червні 2015 року. Це також принесло якусь плутанину в зв'язку з назвою пакета, адже ES6 і ES2015 - це одне і те ж. З виходом цього пакету оновлень комітет прийняв рішення перейти до щорічних оновлень. Тому видання було перейменовано в ES2015, щоб відображати рік релізу. У оновленні ES2015 були зроблені наступні зміни:

- додані стрілочні функції;

- в шаблонних рядках можна оголошувати рядки за допомогою `(зворотних лапок). Шаблонні рядки можуть бути багаторядковими, також можуть інтерполювати;
- let і const - альтернативи var для оголошення змінних. Додана "тимчасова мертва зона";
- ітератор і протокол ітерації тепер визначають спосіб перебору будь-якого об'єкта, а не тільки масивів. Symbol використовується для присвоєння ітератора до будь-якого об'єкта;
- додані функції-генератори. Вони використовують yield для створення послідовності елементів. Функції-генератори можуть використовувати yield * для делегування в іншу функцію генератора, крім цього вони можуть повертати об'єкт генератора, який реалізує обидва протоколи;
- додан Проміс конструктор.

Але нажаль не усі браузери підтримують нові стандарти JavaScript (наприклад браузер FireFox зовсім не подтримує тег module, рис. 18) тож ми використовуэмо спеціальну **jQuery** та **Babel.js** для забезпечення коректної роботи JavaScript в усіх браузерах.

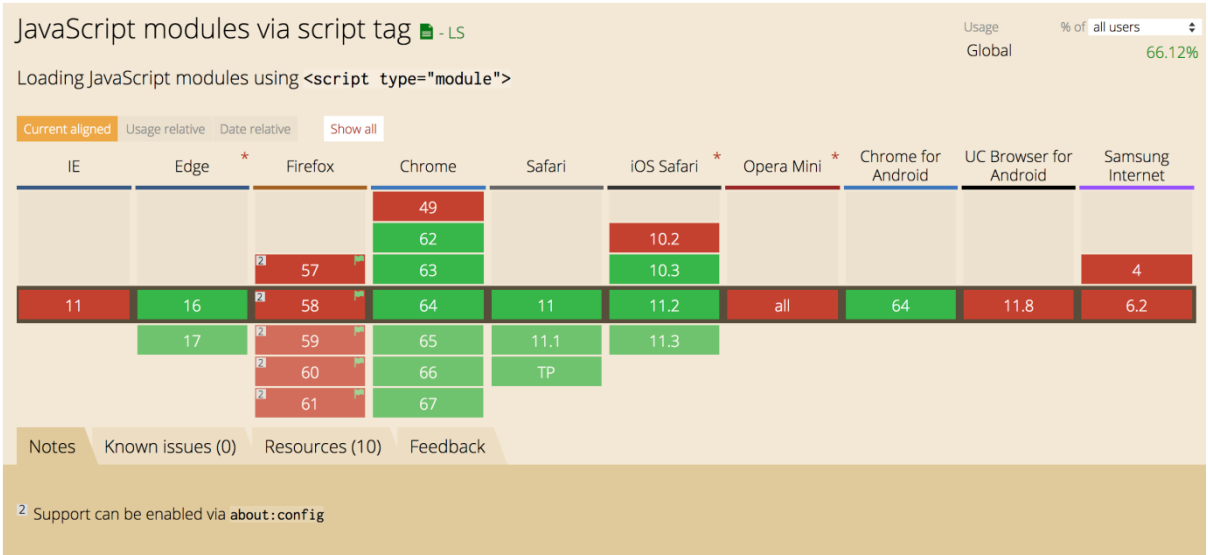


Рис. 18

jQuery - популярна JavaScript-бібліотека з відкрітім вхіднім кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресіґом (John Resig).

Згідно з дослідженнямі организации W3Techs, JQuery вікорістовується понад половиною від міліона найвідвідуванішіх сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка тосі використовується.

jQuery є вільним програмним забезпеченням під ліцензією MIT (до вересня 2012 Було Подвійне Ліцензування під MIT та GNU General Public License Другої Версії)

Найчастіше використоувані функції у проекті:

after() / before()

Ми використовуємо ці два методи коли нам потрібно вставити що-небудь в DOM, але у вас немає належних зачіпок, за допомогою яких можна це зробити; append() або prepend() не дозволяють йти навпростець.

Дві ці функції могли б стати тим, що потрібно. Вони дозволяють вам вставляти елементи в DOM прямо перед, або після обраного елемента таким чином, що новий елемент поміщається на той же рівень.

change()

Метод change() - це оброблювач подій, так само як click() або hover(). Подія change - для текстових областей, полів введення і вікон вибору, і воно спрацює, коли зміниться значення цільового елемента; зауважте, це не те ж саме, що обробники подій focusOut () або blur (), які запускаються,

коли елемент втрачає фокус, незалежно від того, змінено його значення чи ні.

Подія change() бездоганно для валідації даних на стороні клієнта; це набагато краще, ніж blur(), тому що ви не будете заново виконувати перевірку полів, якщо користувач не змінить значення.

Ми використовуємо цей метод для валідації полей форм.

grep()

Ми використовуємо її, щоб фільтрувати масив елементів. Це не метод колекцій jQuery; ми передаємо масив в якості першого параметра і функцію фільтрації як другий параметр. Функція фільтрації приймає два параметри: елемент з масиву і його індекс.

Метод повиннен виконати свою роботу і повернути значення true або false. За замовчуванням, всі елементи, для яких було повернуто значення true, будуть потрапляти в результати фільтрації.

Babel.JS - це транспайлер, переписуючий код розроблений згідно станларту ES-2015 код на попереднього стандарту ES5.

Він складається з двох частин:

- Власне транспайлер, який переписує код.
- Поліфілл, який додає методи Array.from, String.prototype.repeat і інші.

Зазвичай Babel.JS працює на сервері в складі системи збирання JS-коду (наприклад webpack або brunch) і автоматично переписує весь код в ES5. У нашому випадку ми використовуємо систему збірки коду **Gulp.**

Налаштування такої конвертації тривіальна, єдино - потрібно підняти саму систему збирання, а додати до неї Babel легко, плагіни є до будь-якої з них.

Gulp - це інструмент збірки веб-додатків, що дозволяє автоматизувати повторювані завдання, такі як складання і мініфікація CSS- і JS-файлів, запуск тестів, перезавантаження браузера і т.д. Тим самим Gulp прискорює і оптимізує процес веб-розробки.

Gulp побудований на **Node.js**, і файл збірки пишеться на JavaScript. Сам по собі Gulp вміє не дуже багато, але має величезну кількість плагінів, які можна знайти на сторінці зі списком плагінів або просто пошуком на npm. Наприклад, є плагіни для запуску JSHint, компіляції CoffeeScript, запуску тестів і навіть для поновлення номера версії збірки.

Головна перевага Gulp перед Grunt якраз і полягає в використанні потоків. Grunt не використовує потоки, він бере файли, виконує з ними якусь операцію і зберігає, і так для кожної операції. Часте звернення до файлової системи значно уповільнює роботу всієї збірки.

Ми використовуємо Gulp у нашому проєкті для вирішення далі перелічених потреб:

- Валідація та компіляця набору LESS файлів у єдиний файл каскадної таблиці стилей.
- Додавання вендорних префіксів до каскадних таблиць стилей.
- Мініфікація CSS.

- Мініфікація зображень
- Обьєднання усіх файлів JavaScript у эдиний.
- Компіляція JavaScript ES6 у ES5 для забеспечення підтримки коду в усіх браузерах.
- Валідація та мініфікація файлу скриптів.
- Перезбірка проєкту після кожного збереження будь-якого файлу підчас розробки.
- Перезавантаження вікна браузеру підчас роботи над статичними файлами, одразу після перезбірки проекту.

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Засновником платформи є Раян Дал (Ryan Dahl). Платформа node.js перетворила мову JavaScript, що в основному використовувалась в браузерах, на мову загального використання з великою спільнотою розробників.

Node.js характеризується такими властивостями:

- асинхронна однопотокова модель виконання запитів
- неблокуючий ввід/вивід
- система модулів CommonJS

Для керування модулями використовується пакетний менеджер **npm (node package manager)**.

Використовуючи саме цей модуль ми маємо змогу завантажити усі необхідні компоненти для налаштування та коректної роботи з Gulp:

- Rigger

- Rename
- Watch
- Prefixer
- Optimizejs
- Less
- Cssmin
- Rimraf
- Browsersync
- Nitify
- Reload
- Replace
- Mmq
- Git
- Babel

Node.js призначений для відокремленого виконання високопродуктивних мережевих застосунків на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для веб, платформа може використовуватися і для створення звичайних клієнтських і серверних мережевих програм. Для забезпечення виконання JavaScript-коду використовується розроблений компанією Google рушій V8.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів (callback). Як способи мультиплексування з'єднань підтримується

epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libuv, для створення пулу потоків (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що спричиняють блокування, виконуються всередині пула потоків і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки Perl AnyEvent, Ruby Event Machine і Python Twisted, але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у веб-застосунку, що працює в браузері.

Висновки до розділу 2

У цьому розділі ми розглянули усі необхідни інструменти для проектуання, розгортування, реалізації проекту, роботи зі статичними і динамічними файлами, та версіюванням проекту, та автоматизації процессу розробки.

Порівнявши типи управління проектами та системи автоматизації процесів розробки, можливо зробити висновок що платформа Trello підходить для мінімізації етапів розробки веб-сайтів тому що система надає можливість створювати шаблони досок, та задач, тим самим, підготував ці шаблони один раз, ми маємо можливість використовувати їх у проектуванні процесів.

Наприклад, ми маємо шаблон дошки Trello (Рис. 19), де розміщені шаблони задач які містять у собі додактову інформацію залежно від типу задачі (Рис. 20):

- «Новий компонент» - тип задачі використовується для розробки модулів веб-сайту;
- «Вирішити проблему» - використовується для вирішення проблем на веб-сайті, тож містить у собі покрокову інструкцію як вирішувати типові проблеми;
- «Змінити функціонал» - шаблон задачі використовується у випадках коли постає завдання змінини той чи інший функціонал на веб-сайті, таким чином, щоб нічого не зламати;
- «Оновити контент» - кожен раз коли контент на веб-сайті необхідно змінити, нова задача буде зроблена з використанням цього шалону, який у свою чергу містить інструкції по оновленню контенту;
- «Розслідування» - данний тип задачі використовується у випадках коли, щось працьоє не корректно, тож програмісту необхідно з’ясувати обставини та знайти метод виправлення помилки.

Кожна задача мая кнопку «Send to QA», тож коли задача зроблена, розробник клікне по ній та наступний алгоритм буде відпряцьований:

- Менеджери дошки будуть проінформовані, що задача виконана через модуль повідомлень та електрону пошту;
- Задача отримае додаткові перевірочні чек-лісти для программіста, тож він зможе виконати само-перевірку;
- Задача буде переміщена у список «Review».

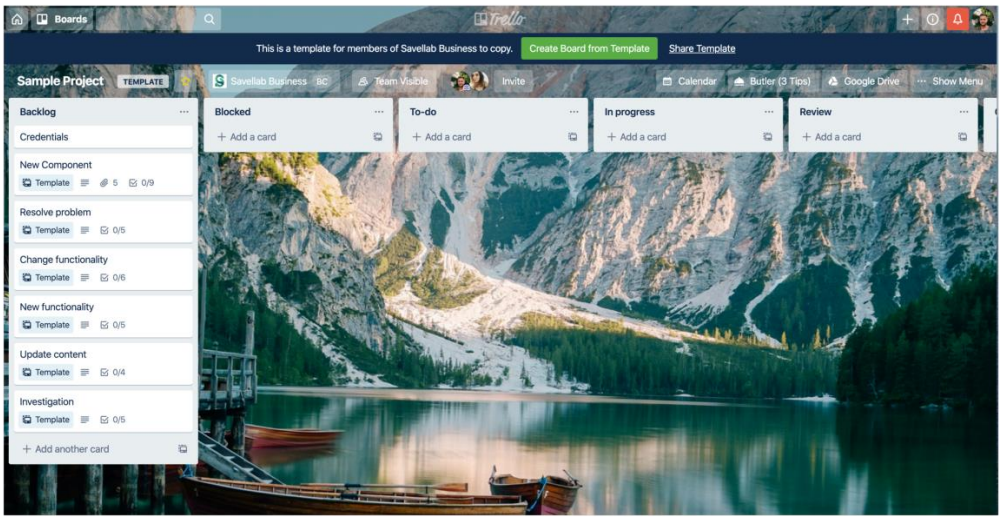


Рис. 19

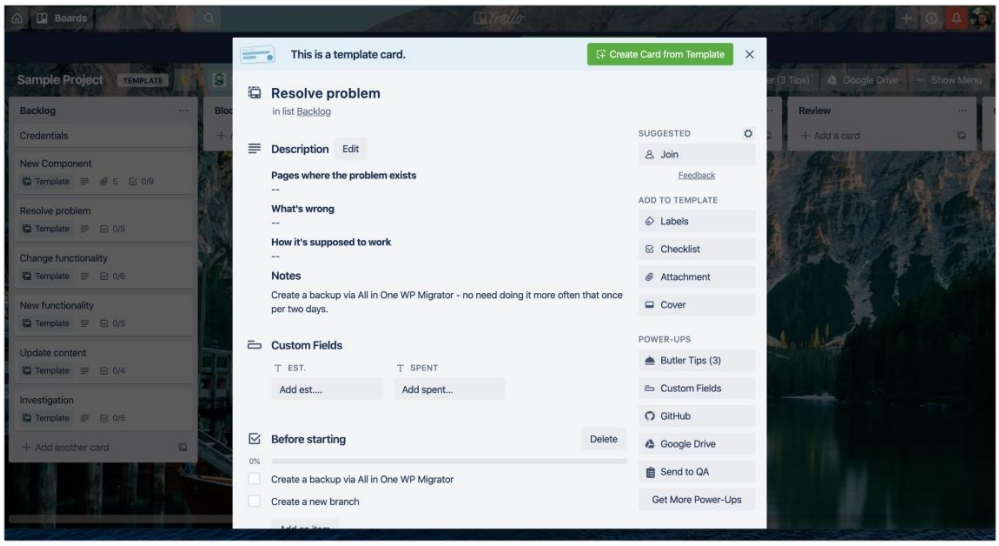


Рис. 20

Аналізуючи інструменти для проектування та розробкки веб-сайтів, у наступному розділі разгянумо технічну реалізацію автоматизації процесів розробки шаблону.

РОЗДІЛ 3: ТЕХНІЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗАЦІЇ РОЗРОБКИ ВЕБ-САЙТУ

1.1. Проектування архітектури Wordpress шаблону

Мета розробки шаблону для Вордпресс – мінімізувати час розробки типових веб-сайтів на туристичну тематику. Тобто, шаблон можливо буде використовувати як основу для подальшого розширення фунціоналу залежно від харатеристик проекту.

У нашому випадку, шаблон повинен буди спроєктованим таким чином, щоб містити наступні модулі

- Обробка вхідного коду статичних файлів та автоматична мініфікація CSS та JavaScript ресурсів;
- Конструктор сторінок, метою якого є автоматизація побудови внутрішніх сторінок веб-сайту за допомогою інтерфейсу зрозумілого людині не знаьомою із програмуванням;
- Центр модифіккації зовнішнього вигляду веб-сайту за допомогою якого можниво змінювати кольори, шрифти, розміри, основних компонентів веб-сайту, таких як, заголовки, параграфи, панелі навігації тощо.

Модуль обробки статичних файлів

Для реалізації автоматичної мініфікації файлів використовуеться Node.js який був згаданний у попередньому розділі.

Але важливо підключити ресурси таким чином, щоб сменшити час завантаження веб-сторінки, а також автоматизувати процес підключення додаткових скриптів на веб-сайт.

Для виконання вищезгаданних вимог необхідно зробити наступне

- Збирати усі JavaScript файли у єдиний файл та підключати його за допомогою атрибуту defer;
- CSS ресурси повинні бути додані на сторінку задопомогою inline тегу style таким чином, щоб браузеру не потрібно було робити додатовий запит на сервер для отримання стилей;
- Провадити API для додавання нових ресурсів, якщо буде запотрібно у процесі розробки.

Тож, для виконання цих вимог, розробимо класс “Assets” у якому за допомогою зазначених на рисунку 1 методів досягнемо необхідного результату.

- __construct – метод конструктору який відпряцьовується на момент ініціалізації веб-сайту. Данний метод містить у собі Wordpress хуки за допомогою яких можливо модифікувати фунціонал платформи, у нашому випадку, обробити статичні ресурси. Тож усі методи наведені нижще насправді використовуються у хуках;
- deregister_assets – метод для відключення усіх скриптів які Wordpress підключає за завмовчуванням;
- create_assets_bundles – метод відповідальний за оновлення статичних ресурсів та збереження іх у єдиний файл;
- check_theme_version – метод який відповідає за виклик функції create_assets_bundles тільки тоді коли шаблон був оновленний;
- enqueue_assets – метод для підключення вихідного коду статичних ресурсів;

- add_async_attribute – метод для додавання атрибуту defer у тег script
- concat_assets – метод у якому безпосередньо відбувається компіляція усіх ресурсів у єдиний файл;
- include_css_libraries – API метод для підключення CSS файлів, який приймає у себе масив посилань на ресурси стилей;
- include_js_libraries– API метод для підключення JavaScript ресурсів, який приймає у себе масив посилань на ресурси скриптів.

Конструктор сторінок

Для реалізації функціоналу конструктору сторінок використовується вбудований у Wordpress модуль під назвою Gutenberg який базується на блоках.

Одне, що відрізняє WordPress від інших систем, це те, що воно дозволяє користувачам створювати максимально багатий макет публікацій, наскільки вони можуть собі уявити, - поки вони знають HTML та CSS та будують власну тему.

Гутенберг переробляє редактор на інструмент, який дозволяє користувачам писати багаті пости та створювати красиві макети за кілька кліків - технічних знань не потрібно.

На Рис. 21 представленний зовнішній вигляд конструктору.

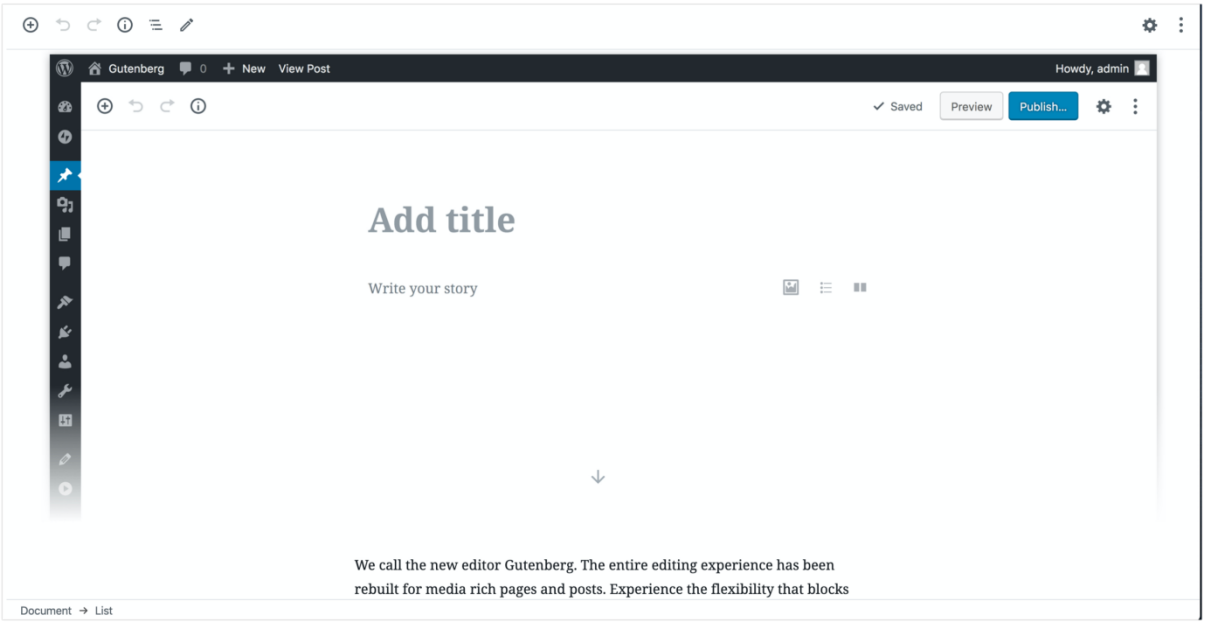


Рис. 21

Все на веб-сайті WordPress стає блоком: текст, зображення, галереї, віджети, шорт-коди та навіть шматки спеціального HTML, додані плагінами чи іншим способом. Користувачам доведеться вивчити лише один інтерфейс - блок-інтерфейс.

Усі блоки створюються рівними. Всі вони живуть в одному інтерфейсі вставки. Рецензія, пошук, вкладки та групування забезпечують доступність блоків, які використовуються найчастіше.

Перетягування є другорядним. Для більшої доступності та сумісності платформи взаємодія перетягування використовується як додаткове покращення поверх явних дій, таких як клік, вкладка та пробіл.

Ключові місця заповнювачів. Якщо блок може мати нейтральний стан заповнення, він повинен. Блок заповнення зображень показує кнопку для відкриття медіатеки, а блок заповнення тексту відображає запит на запис. Використовуючи заповнювачі, ми можемо заздалегідь визначити редаговані макети, тому всі користувачі повинні зробити це заповнити пробіли.

На Рис. 22 наведені блоки які доступні одразу після встановлення Wordpress платформи.

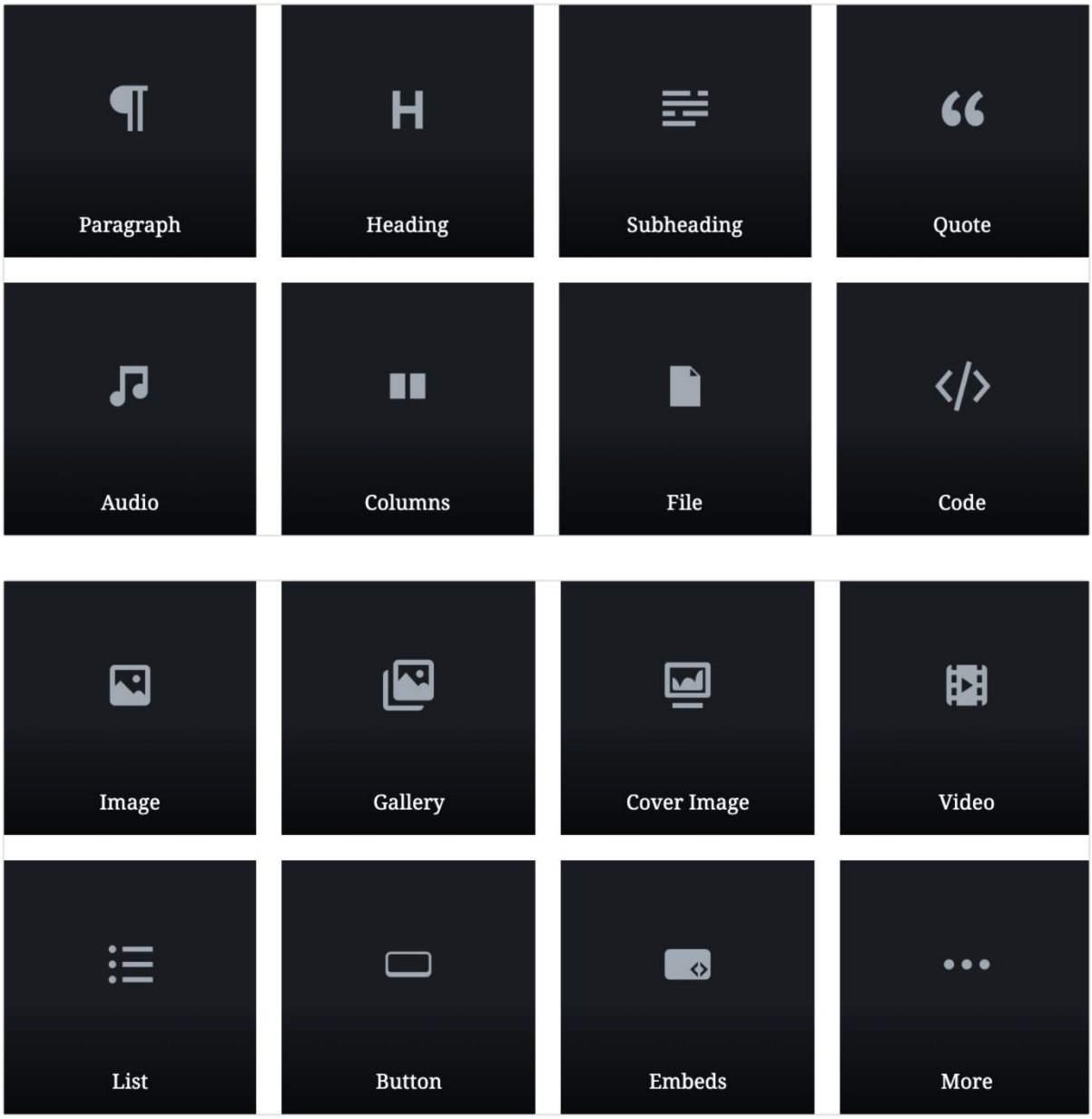


Рис. 22

Пряма маніпуляція інтуїтивно зрозуміла. Блок-інтерфейс дозволяє користувачам маніпулювати вмістом безпосередньо на сторінці. Автори плагінів та тем підтримуватимуть та розширюватимуть цей досвід, будуючи власні власні блоки.

Редагування коду не повинно бути необхідним для налаштування. Налаштування традиційно вимагає складної розмітки, а складну розмітку легко зламати. З Гутенбергом налаштування стає більш інтуїтивно зрозумілим та безпечнішим.

Розробник зможе надати користувацькі блоки, які безпосередньо надають частини макета (наприклад, сітка з трьох стовпців) та чітко вказати, що може безпосередньо редагуватися користувачем. Це означає, що користувач може оновлювати текст, міняти зображеннями, зменшувати кількість стовпців, не вимагаючи запитання розробника чи турбуючись про те, щоб зламати речі.

В основі Гутенберга лежить концепція блоку. З технічної точки зору, блоки обидва підвищують рівень абстрагування від одного документа до колекції змістовних елементів, і вони замінюють неоднозначність - властиву HTML - явною структурою.

З точки зору користувача, блоки дозволяють безпосередньо додавати будь-який вміст, засоби масової інформації чи функціональні можливості на свій сайт більш послідовно та зручно. Кнопка «Додати блок» надає користувачеві доступ до всієї бібліотеки параметрів, що знаходяться в одному місці, замість того, щоб шукати через меню чи знати шорткоди.

Але найголовніше, що Гутенберг побудований за принципом прямої маніпуляції, це означає, що основні параметри того, як відображається елемент, керуються в контексті самого блоку.

Це є великим зрушенням від традиційної моделі WordPress, де параметри, які часто закопувались глибоко в шари меню навігації, контролювали елементи на сторінці за допомогою непрямих механізмів.

Так, наприклад, користувач може додати зображення, написати його підписи, змінити його ширину та макет, додати посилання навколо нього,

все з інтерфейсу блоку на полотні. Цей же принцип повинен застосовуватися і до більш складних блоків, як, наприклад, "навігаційне меню", при цьому користувач може додавати, редагувати, переміщувати та завершувати повну презентацію своєї навігації.

Користувачам потрібно вивчити лише один інтерфейс - блок - для додавання та редагування всього на своєму сайті. Користувачі не повинні писати короткі коди, користувацькі HTML або розуміти приховані механізми вбудовування вмісту.

Таким чином процес розробки веб-сайту стає актоматизованим, а більшість роботи може бути виконана без досвіду програмування. На Рис. 23 розглянемо архітектуру блоку:

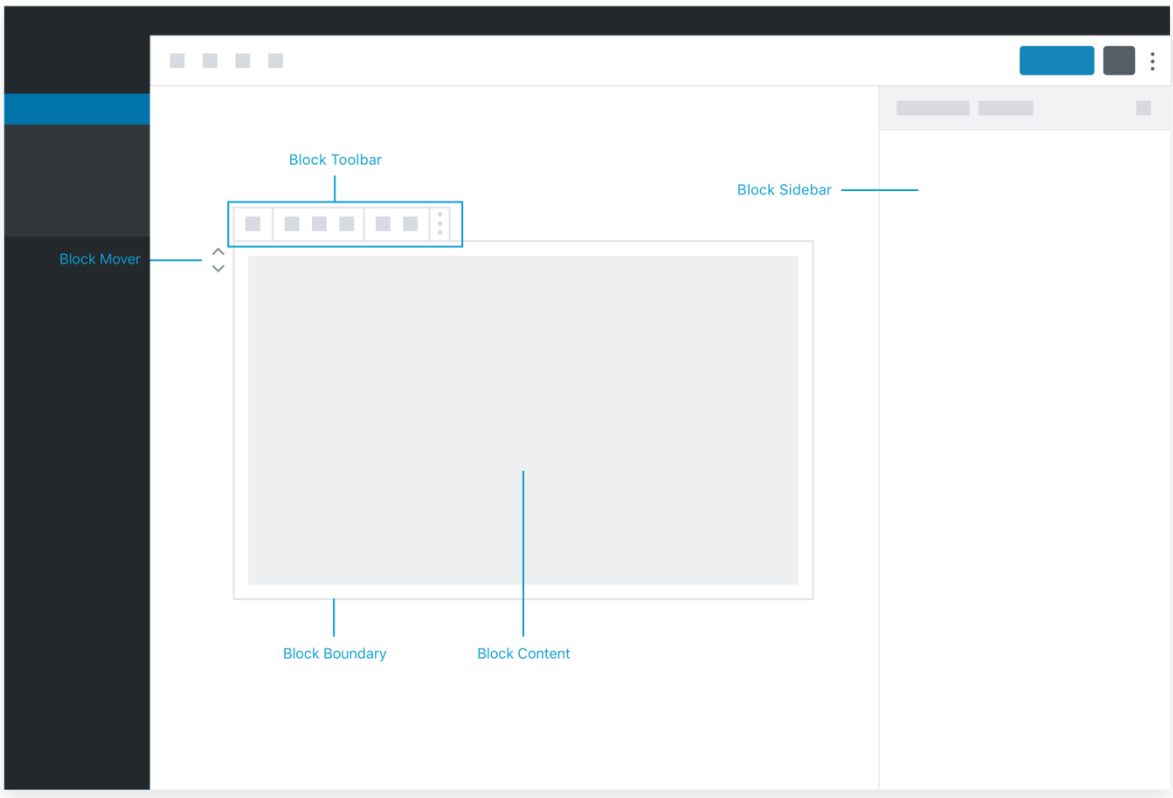


Рис. 23

Основним інтерфейсом для блоку є область вмісту блоку. Вміст заповнювачів в області вмісту блоку може розглядатися як керівництво або інтерфейс для того, щоб користувачі виконували набір інструкцій або «заповнювали пробіли» (детальніше про заповнювачі заповнення пізніше).

Оскільки область вмісту відображає те, що насправді з'явиться на сайті, взаємодія тут виявляється найбільш близькою до принципу прямої маніпуляції і буде найбільш інтуїтивно зрозумілою для користувача. Це слід розглядати як основний інтерфейс для додавання та маніпулювання вмістом та коригування його відображення.

На Рис. 24 наведен зовнішній вигляд модулю додавання блоків.

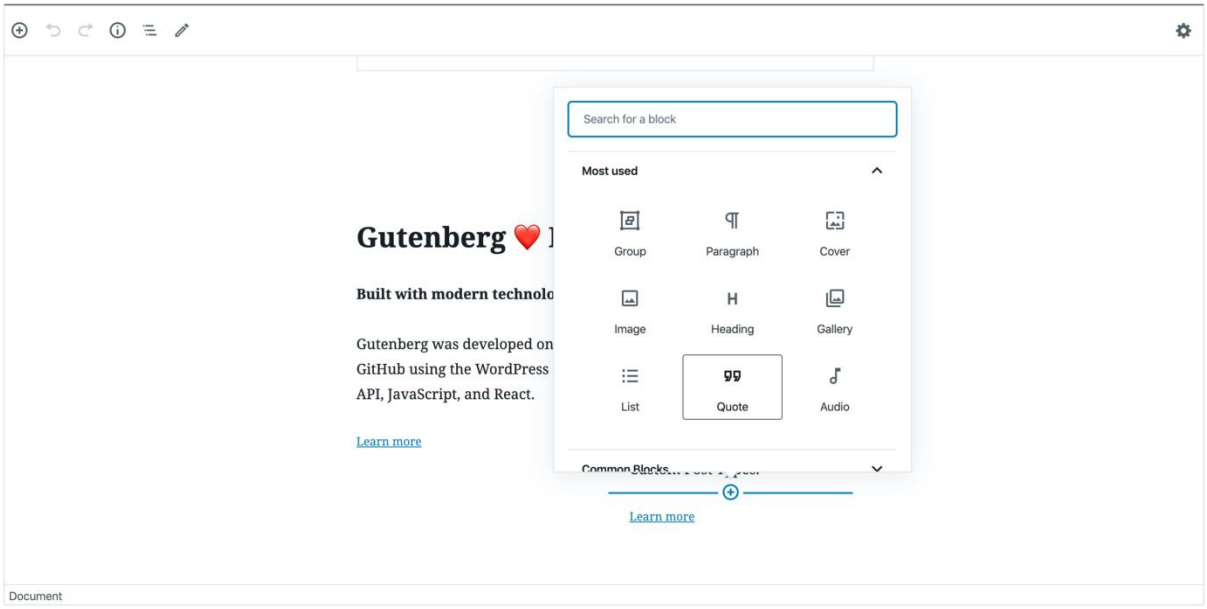


Рис. 24

Блокова панель інструментів - це місце для критичних параметрів, які не можна включити в інтерфейс заповнення. Основні параметри блоку не завжди матимуть сенс у контексті інтерфейсу для заповнення / вмісту. В

якості другого варіанту параметри, які є критичними для функціональності блоку, можуть жити на панелі інструментів блоку. Блокова панель інструментів - це один крок, що видаляється від прямої маніпуляції, але все ще є дуже контекстуальною і видимою для всіх розмірів екрана, тому це чудова додаткова можливість.

Центр модифіккації зовнішнього вигляду веб-сайту

Посилаючись на той факт, що різні веб-сайти мають відмінні риси у зовнішньому вигляді, реалізуємо можливість змінювати дизайн інтерфейсу сторінок.

Модуль відповідальний за виконання наступних функцій:

- надання інтерфейсу для змінення зовнішнього вигляду веб-сторінок
- генерація CSS файлу стилей
- можливість зберігання окремих рис, таких як кольори, та шрифти. Наприклад, можливо зберегти значення колюру у панері констант, та використовувати його для стилізування тексту або блоів.

Таким чином, розробивши палітру кольорів у шаблоні, можливо автоматизувати процес стилізації веб-сайту змінюючи головні змінні из панелі констант.

На Рис. 25 зображений зовнішній вигляд панелі констант кольорів.

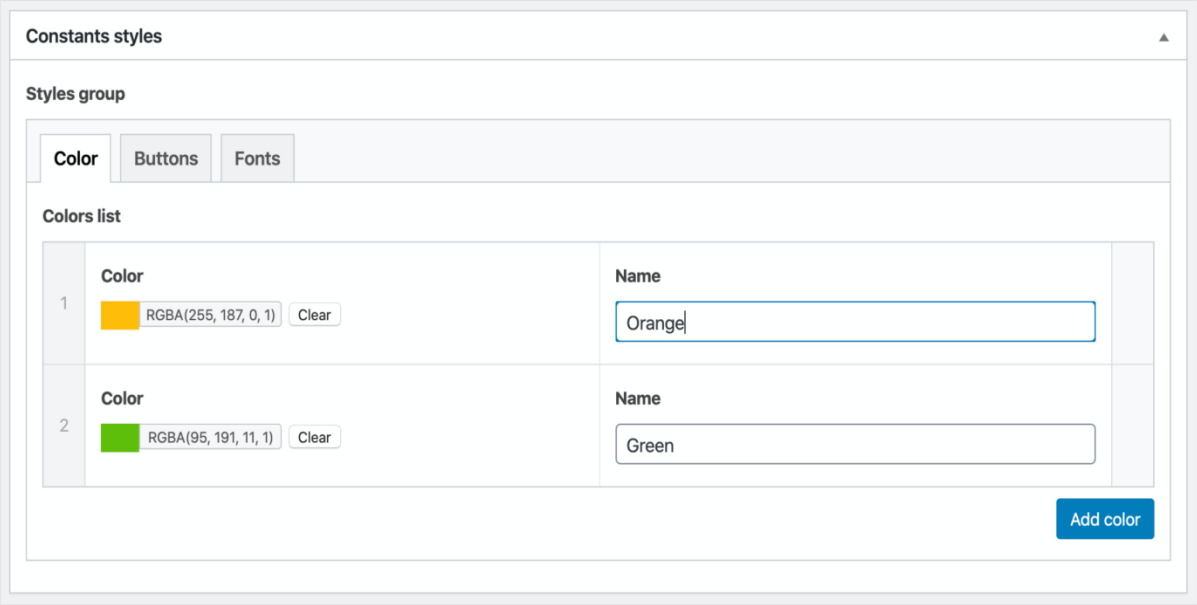


Рис. 25

Також розглянемо Рис. 226 на якому зображенно приклад використання збереженного кольору у контексті стилізації елементу посилання. Крім того, елемент посилання має наступні риси доступні для модифікації:

- стиль шртифту
- розмір шрифту
- висота базової лінії шрифту
- відстань між буквами
- декорації тексту
- трансформація тексту, наприклад можливіть зробити текст великими літерами
- вирівнювання тексту
- тип кольору, який може бути довільним або константою

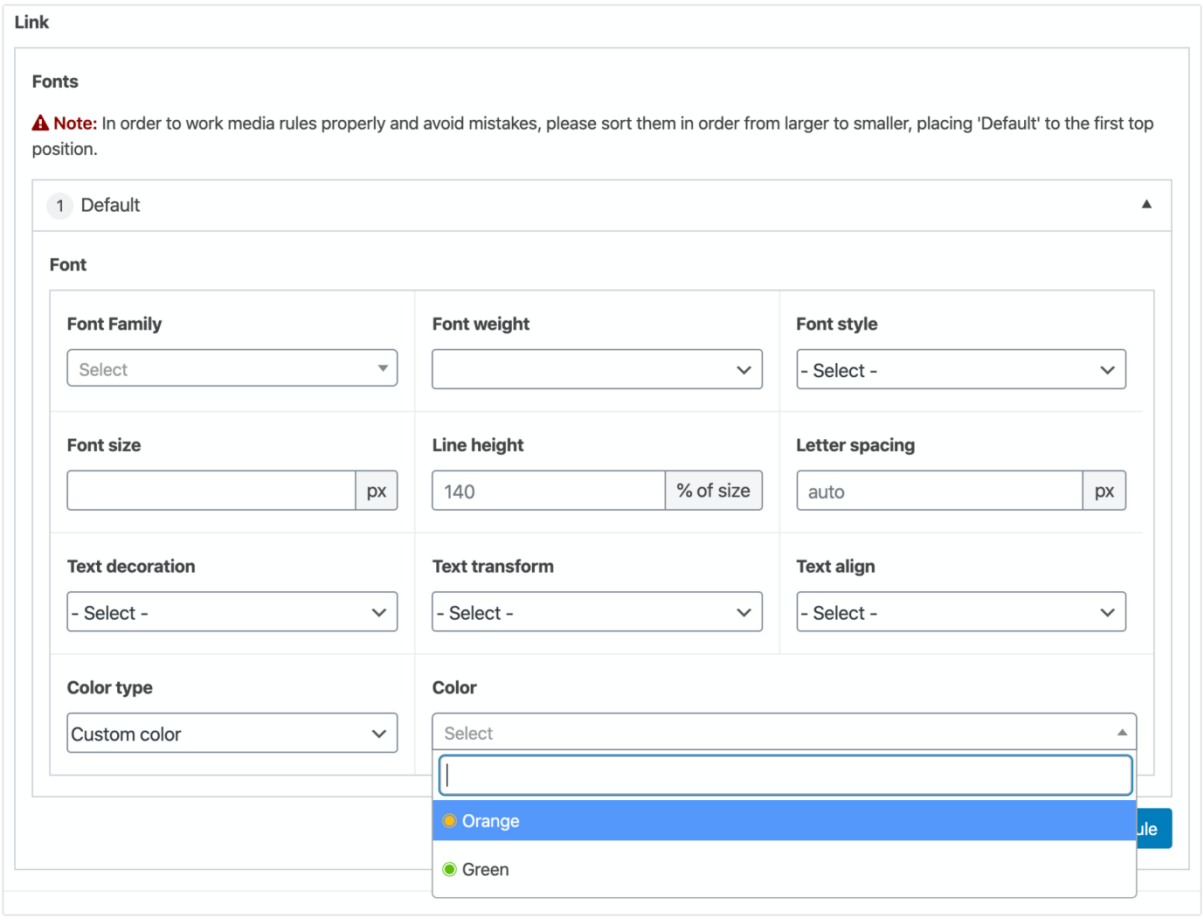


Рис. 226

Загалом усі елементи веб-сайту можуть бути стилізованими за допомогою розгляненого модулю, а стилі повторно використаними у різних місцях веб-сайту, таким чином немає необхідності у написанні CSS коду для кожного, тож цей процес повністю автоматизованно.

1.2. Розширення можливостей довільних полей для записів

WordPress дозволяє авторам додавати довільні поля до записів. Вміст цих полів називають meta-data. Поля можуть включати деяку кількість додаткової інформації.

За допомогою програмування можна додати до довільних полів більш складні функції. Наприклад, вони можуть містити "термін придатності" записи.

Дані зберігаються парами - ключ / значення. Ключ - це назва довільного поля. А значення - це вміст поля, яке буде індивідуальним для кожного запису.

Один і той же ключ може бути використаний в запису кілька разів. Наприклад, якщо ви читаєте дві різні книги (технічне керівництво на роботі і фантастику будинку), ви могли б створити поле "Читаю" і використовувати його двічі в одному записі (по одному разу для кожної книги).

Advanced Custom Fields плагін для WordPress робить роботу з довільними полями легкої і швидкої. Довільні поля вбудовані в WP, але вони досить незручні в налаштуванні і не зручні для користувача.

У Групах полів зберігаються довільні поля, умови відображення і візуальні налаштування. Кожній групі можна вказати де вона повинна виводиться на основі маси налаштувань.

Усередині групи полів можна створювати окремі довільні поля. Наприклад, створюємо групу полів "Схожі туристичні тури" і в ній 2 поля "Назва туру" і "Зображення". Кожне поле має безліч параметрів, які змінюють тип (текст, зображення, редактор wysiwyg, списки і т.д.), значення за замовчуванням, залежно та багато іншого.

Дозволяє налаштувати візуальне відображення для групи полів. Також можна приховати непотрібні елементи Інтрфейс, щоб не відволікати користувача. Наприклад, можна приховати стандартне вікно редагування і

поставити туди своє. Розвантажує робочий простір і зробити більш зручним процес роботи. На рис. 27 зображений проклад Полів відсортированих по табам, що значно розвантажує простір.

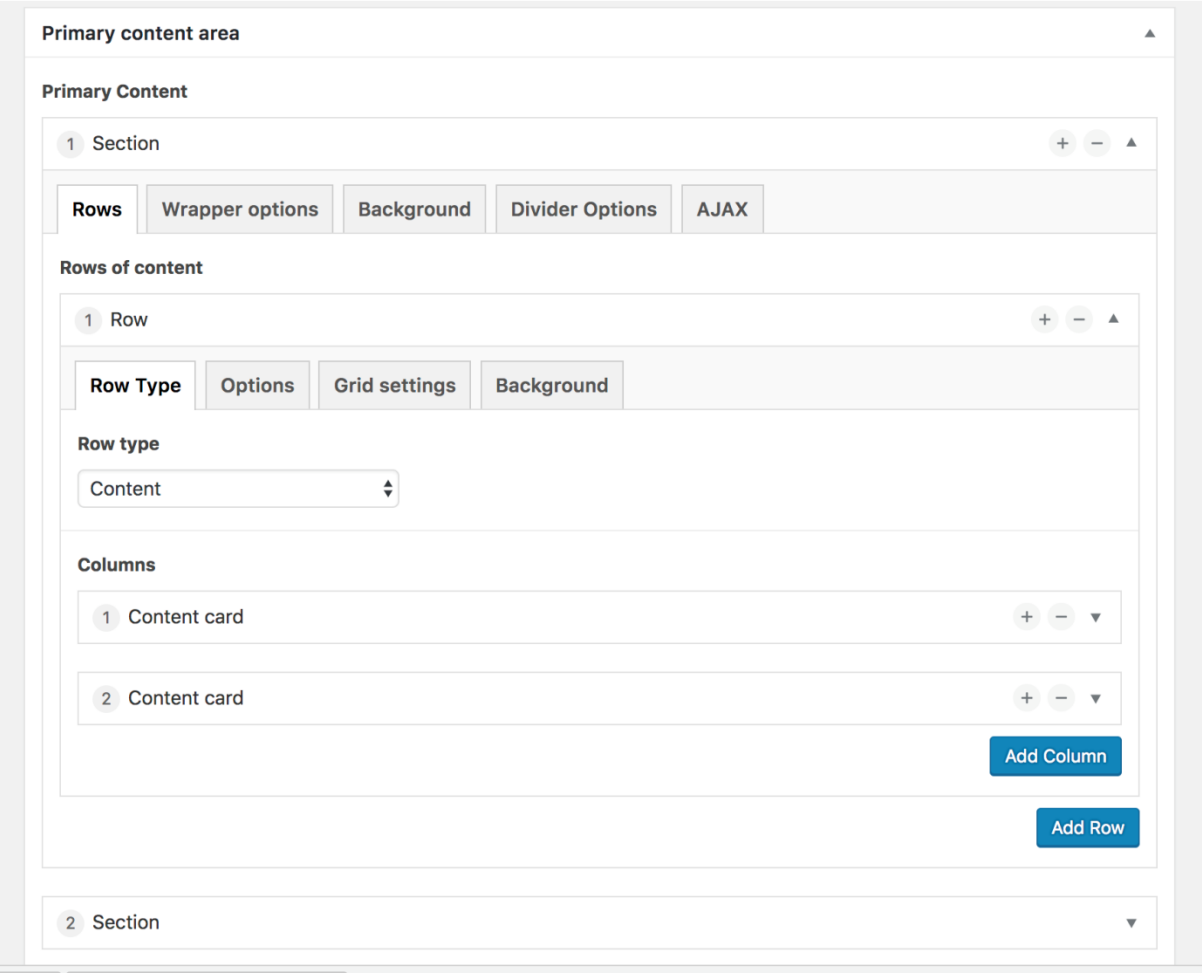


Рис. 27

У ACF вбудовано безліч потужних і корисних функцій для роботи з даними. Завдяки цьому код стає компактніше і розумніші.

У розробці шаблону теми ми використовуємо наступні типи полів:

- Text
- Textarea
- Range

- Checkbox
- Radio button
- Select
- True/false
- Button group
- Radio button
- File
- Gallery
- Image
- oEmbed
- Wysiwyg Editor
- Color Picker
- Date Picer
- Date Time picker
- Google map
- Time picker
- Accordion
- Clone
- Flexible content
- Group
- Repeater
- Tab
- Link
- Page link
- Post object
- Relationship
- Taxonomy

Text (довільне поле Текст)

Надаэ можливість використовувати у записах короткий текст з додатковою інформацією. На боці панелі адміністратора, данне полє являється тегом INPUT, а на клієнтській – може будти викликано за допомогою PHP функції get_field де первим аргументом виступає назва довільного поля, а другим – унікальний ідентифікатор запису.

Ми використовуємо дане поле для надання додаткової та короткої інформації о туристичних пропозиціях, наприклад, додаткові вимоги.

Повертає строкове значення.

Textarea (довільне поле Область тексту)

Має абсолютно такі ж самі можливості як довільне поле Текст, але має призначення для більних обсягів тексту та використорує тег TEXTAREA на боці панелі адміністратора.

Наш шаблон теми використовує Область тексту для надання можливості адміністратору додавати опис туристичних пропозицій.

Повертає значення у строковому типі.

Range (довільне поле Діапазон значеннь)

Надає можливість задавати діапазон значень на боці панелі адміністратора, та повертає значення діапазону на боці клієнта.

Повертае масив значень.

Checkbox (довільне поле Відмітка)

На боці редагування запису, представляє собою набір єлементів INPUT TYPE CHECKBOX та надає можливість повертати індивідуальні значення на клієнтському боці. Наприклад, для якогось туру, ми можемо відмітити Відмітку «Акційний тур», та за допомогою PHP функції get_field опрацювати інформацію.

Повертає масив значень відмічених на боці редагування запису.

Radio button (довільне поле Вибір)

На боці редагування запису, представляє собою набір єлементів INPUT TYPE RADIOBOX та надає можливість повертати одне значення із списку доступних на боці редагування сторінки.

Повертае значення у строковому типі данних.

Select (довільне поле Вибірка)

На боці редагування запису, представляє собою набір єлементів SELECT. Надає можливість обрати один або декілька варіантів із списку та використовувати їх на клієнтському боці.

Повертає масив обраних значень.

True/false (довільне поле Перемикач)

Ми використовуємо данний тип довільного поля для роботи з данними у булевому типі.

Повертает булеве ВІРНО або НЕВІРНО.

Button group (довільне поле Група кнопок)

Виконує тіж самі функції що довільне поле Вибір, але предствавляє собою більш зручний вигляд на боці редагування запису.

Повертає строкове значення.

File (довільне поле Файл)

Надає можливість прикріпити додатковий файл на боці редагування запису та використовувати його на боці клієнту. Залежно від налаштувань, повертає або лінк на файл у строковому типі, або масив данних файлу.

Gallery (довільне поле Галерея)

Незважаючи на те, що Вордпресс надає можливість створювати галереї у текстовому редувачі Wysiwyg, данне довільне поле надає більш гнучкіші можливості з роботою вихідними данними.

Повертає колекцію масивів зображень галереї.

Image (довільне поле Зображення)

Перш за все, Вордпрес має функціонал для додавання зображення до кождого запису, а також додавання зображень у Wysiwyg редактор, але у розробці гнучких шаблонів тем для Ворпдрессу є необхідність у додаткових ображеннях на сторінці.

Залежно від нашалтувань повертає масив данних зображення або його ідентифікаційний номер.

oEmbed (довільне поле Відео зі стороннього сервісу)

Надає можливіть використовуючи посилання на відео, використовувати його на клієнтському боці за допомогою спеціальної функції get_field. Здебільного використовується на сторінках туристичних продуктів для більш датальної презентації пропозиції. Повертае єлемент IFRAME у строковому вигляді.

На рис. 28 зображений зовнышный вигляд довільних полей: Текст, Область тексту, Діапазон значень, Відмітка, Вибір, Вибірка, Перемикач, Група кнопок, Файл, Галерея, Зображення, Відео зі стороннього сервісу.

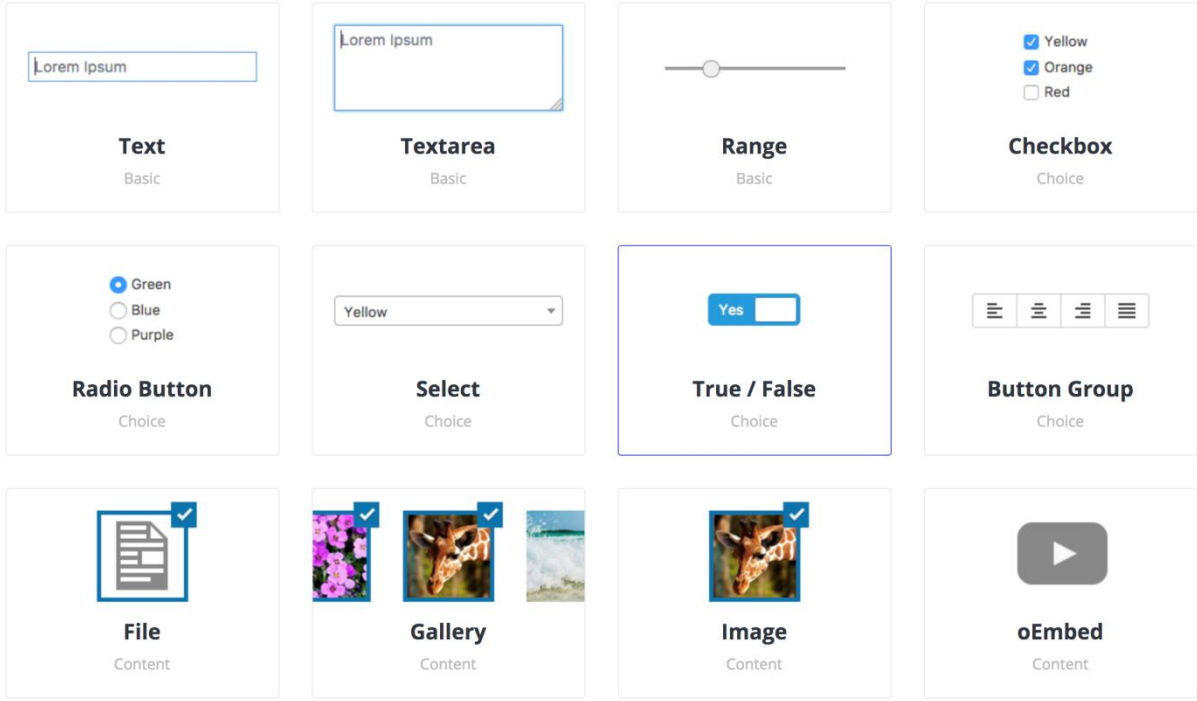


Рис. 28

Wysiwyg Editor (довільне поле Редактор тексту)

Виконує таку саму функцію як Область тексту, але надає можливість його стилізувати та автоматично додає усі необхідні теги.

Данний тип текстовго поля використовується Вордпрессом для роботи із записами, але при розробці власного щшаблону, постає необхідність у використанні Редактору тексту у декількох місцях сторінки. Повертає дані у строковому вигляді.

Color Picker (довільне поле Налаштунок кольору)

Використовується для зручного задання кольору на сторінці редагування запису.

Ми використовуємо у тих випадках коли необхідно зробити колір єлементу сторінки зручним для редагування.

Повертае колір у форматі HEX та у строковому типі даних.

Date / Time / Date Time picker (довільне поле Налаштування дати та часу)

Надає можливіть задати дату та час у зручному вигляді на боці редагування сторінки, використовуючи бібліотеку jQuery.

Розробляючи тему для сайтів на туристичну тематику, існує необхідність за використанні данного типу поля на сторінках турів, де представляється загальна інформація, час та дата туру.

Google map (довільне поле Мапа)

Дане поле надає можливіть задати координати точки на мапі використовуючи лише адресу та візуальний інтерфейс. Повертае масив данних обраної точки координат.

Accordion, Group, Tab (довільні поля Групування)

Поля використовуються для гропування інших довільних полей на сторінці редагування запису. Нічого не повертає на боці клієнта.

Clone (додаткове поле Клон)

Дозволяє групувати поля, але надає на від мину від довільних полей Групування, Клон повертаґ масив значень дочірніх довільних полей.

Repeater / Flexible content (довільні поля Повторювання та Гнучкий контент)

Надає змогу розробляти групи довільних полей та копіювати їх. Повертає колекцію масивів груп довільних полей.

На рис. 29 представленні довільні поля: Редактор тексту, Налаштунок кольору, поля Налаштування дати та часу, Мапа, Групування, Клон, Повторювання та Гнучкий контент.

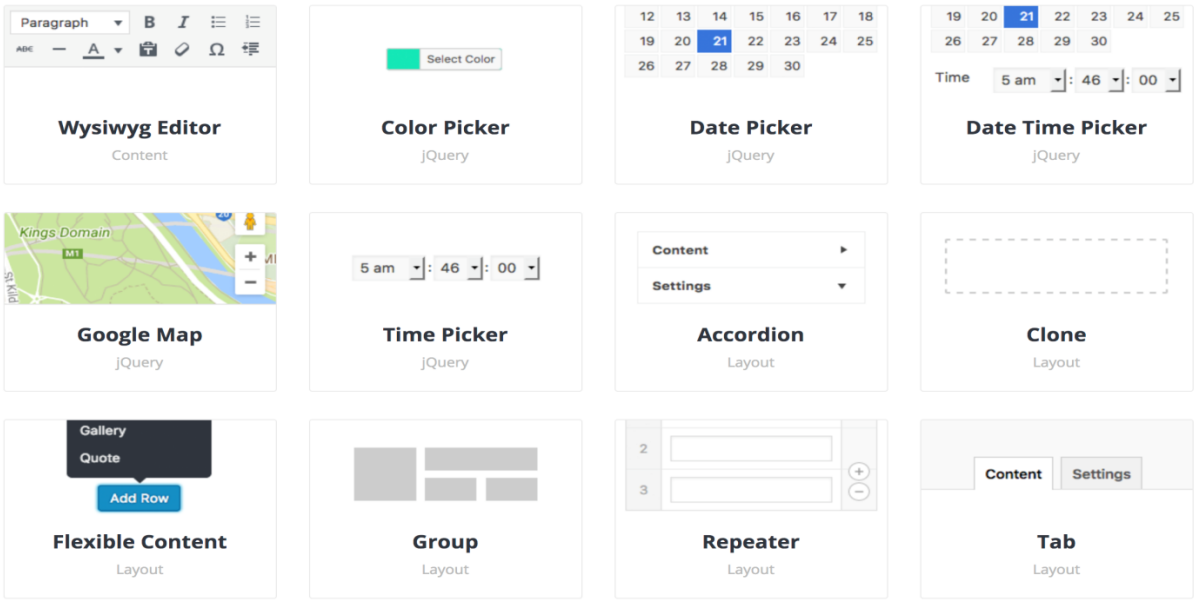


Рис. 29

Link (довільне поле Посилання)

Надає змогу задавати посилання на інший ресурс, також валідує значення на боці редагування запису що запобігає помилкам у написанні посилань.

Повертае посилання у строковому вигляді.

Page link (довільне поле Посилання на запис)

Дає змогу обрати запис посилання на який буде повернено на боці клієнта.

Повертае посилання на запис веб-ресурсу у строковому вигляді.

Post object (довільне поле Об’єкт запису)

На боці редагування сторінки, надає змогу обрати інший запис із списку доступних та використовувати масив його даних на боці клієнта.

Повертає масив данних запису.

Relationship (довільне поле Зв’язки)

Надає гнучкий інтерфейс для повьязаня статей між собою на боці редагування запису.

Повертає колекцію мамисів даних про повьязані записи.

Taxonomy (довільне поле Таксономія)

Надає можливість обрати список категорій, тегів тощо на боці панелі адміністратора, та обробити їх на у шаблонах сторінок.

Повертає масив даних.

На рис. 30 зображений зовнішній вигляд довільних полей:Посилання, Посилання на запис, Об’єкт запису, Зв’язки та Таксономія.

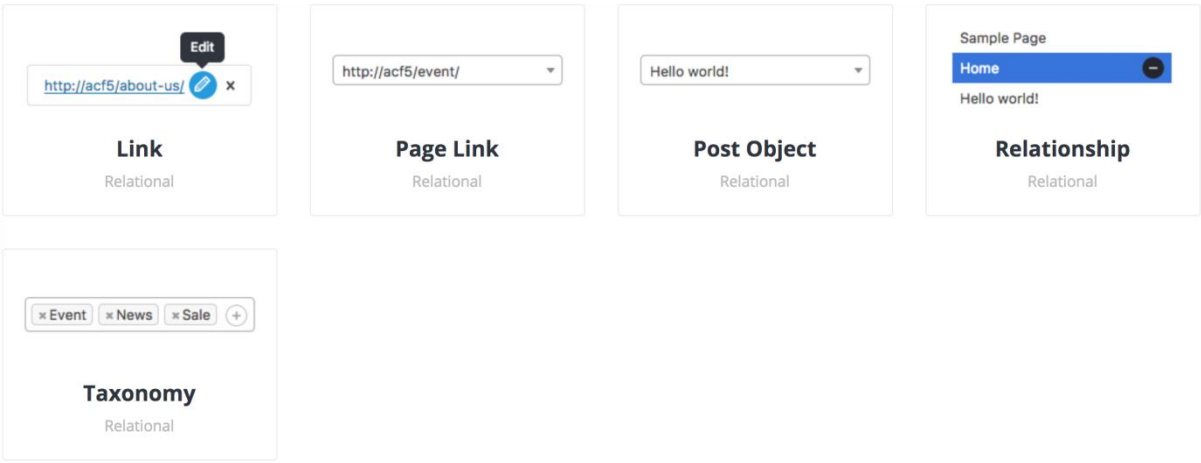


Рис. 30

Основні функції які ми використовуємо для розробки власнного шаблону за допомогою ACF PRO

- get_field($selector, $post_id) - Повертає значення певного поля
- the_field($selector, $post_id) - Відображає значення певного поля
- get_field_object($selector, $post_id) - Повертає параметри (масив) певного поля
- get_fields($post_id) - Повертає масив значень (name => value) для певної публікації
- get_field_objects($post_id) - Повертає масив налаштувань поля (name => field) для певної публікації
- have_rows($field_name, $post_id) - Ця функція використовується для циклічності значення ретранслятора або гнучкого поля вмісту
- get_sub_field($sub_field_name) - Повертає значення певного підполя.
- the_sub_field($sub_field_name) - Відображає значення певного підполя.

- get_sub_field_object($sub_field_name) - Повертає масив даних поля (поле об'єкта + значення) для певного підполя.
- get_row() - Повернення масиву, що містить всі значення підполе для поточного рядка.
- get_row_index() - Повертає поточний індекс рядка в циклі has_rows ()
- get_row_layout() - Повертає поточну назву макета рядка у циклі if_rows ()
- update_field($selector, $value) - Оновлює значення поля.
- delete_field($selector) - Видаляє значення поля.
- update_sub_field($selector, $value) - Оновлює значення підполя.
- add_row($selector, $value) - Додає новий рядок даних до існуючого поля повторювача / гнучкого вмісту вмісту.
- update_row($selector, $row, $value) - Оновлює рядок даних для існуючого значення поля повторювача / гнучкого вмісту.
- delete_row($selector, $row) - Видаляє ряд даних з існуючого значення поля повторювача / гнучкого вмісту.
- add_sub_row($selector, $i, $row) - Додає рядок даних для існуючого підполя значення значення ретранслятора / гнучкого вмісту.
- update_sub_row($selector, $i, $row) - Оновлює рядок даних для існуючого підполя поля значення ретранслятора / гнучкого вмісту.
- delete_sub_row($selector, $i) - Видаляє рядок даних з існуючого значущості підполя ретранслятора / гнучкого вмісту.
- acf_add_options_page($args) - Додайте глобальну сторінку параметрів до інформаційної панелі WP
- acf_add_options_sub_page($page) - Додайте глобальну підсторінку варіантів до інформаційної панелі WP
- acf_form_head() - Перевіряє та зберігає дані, подані з acf_form ().
- acf_form() - Створює передню кінцеву форму.

- acf_register_form() - Реєструє передню кінцеву форму.
- Shortcode [acf field=”field_name”] - Використовується в редакторі вмісту, щоб відобразити значення спеціального поля

1.3. Налаштування гнучкого інтерфейсу роботи із формами

Gravity Forms vs Ninja Forms: особливості

Обидві розширення дозволяють створювати необмежену кількість форм для використання на вашому сайті. Також немає обмеження на обсяг відправлених даних, прийнятих формою. Ці вхідні дані можуть зберігатися в базі даних WordPress. Доступ до них можливий з адміністративної панелі. Також їх можна розсилити на будь-яку кількість електронних адрес. Оба плагіна надають можливість експортувати ці дані у файли різних форматів, включаючи CSV, PDF, та Excel. Крім того, якщо ви хочете показати ці дані більш цікавим способом, сервіс GravityView підходить вам більше.

Типи полей форми

І Ninja Forms, і Gravity Forms дозволяють створювати форми з необхідним набором полів, застосовують правильність полів, а також використовують розрахунки, створені на основі інформації, введеної користувачем. Крім того, доступна опція, що дозволяє створювати умні форми, здатні визначити, які поля потрібно відображати за ступенем заповнення форми користувачем. Вам також буде доступна захист від спаму, яка надається у формі захисних питань.

Типи форм

Форми Ninja і формати Gravity надають нам можливість створювати різноманітні типи форм. Захищені форми, які доступні лише для деяких

користувачів або тільки в певних умовах, багатосторінкові форми або опитування - це лише мала частина того, що можуть створювати ці плагіни. Ви також можете отримувати записи або інші види контент для вашої сторінки, відправлені користувачами через форми, створені Gravity Forms і Ninja Forms. Такий постінг з користувацької частини сайту приносить користь тим власникам сайтів, які привітні створеному користувачами контенту і хочуть зберегти їм час, позбавивши користувачів від необхідності відправляти контент через інтерфейс WordPress-адмінки.

Інтеграція форм

Оба плагіни прекрасно реалізували можливість інтеграції з іншими службами. Включена інтеграція з найпопулярнішими сервісами електронної пошти, платіжними шлюзами, CRM, службами оповіщення, платформами електронної комерції та програмами керування проектами.

Коли справа доходить до набору функцій, тут немає плагіна переможця. Але, так як кожна плагіна доступна в варіаціях, наприклад, є різні тарифні плани, можливість використовувати або не використовувати додатки, то види, створені вами форми буде залежати від того, скільки грошей ви готові заплатити.

Якщо говорити про ті функції, які доступні в одному, і відсутні в другому плагіні, то результати такі. В плагіні Ninja Форми форми з плануванням доступні тільки в певний проміжок часу. У формі Gravity відсутній можем відкрити ваші форми в модальному лайтбоксі-вікні.

Ядро плагіна Ninja Forms, яке можна використовувати безкоштовно, містить безліч великолепних функцій, які дозволяють додати форму зворотного зв'язку для вашої сторінки. Так що, якщо ви шукаєте

безкоштовний і досить потужний плагін для створення форм для вашого WordPress-сайту, Ninja Forms вас не розішить.

Порівняння користувацьких інтерфейсів конструкторів форм.

І в Gravity Forms, і у Ninja Forms використовується інтерфейс оформлений у стилі ядрового інтерфейсу WordPress. Це означає, що в адмінній частині інтерфейси обох плагінів з'єднуються з інтерфейсом самого WordPress. Таким чином, вони почуваються швидше, як звукорядні елементи, а не як сторонні додатки.

Додати поле в форму можна просто, натиснувши на необхідний бокс, що знаходиться в плаваючої панелі. Оба плагіна пропонують інтерфейс "драг-н-дроп" для постановки полів у потрібні позиції.

На рис. 31 зображений інтерфейс Gravity Forms який ми використовуємо у нашому шаблоні.

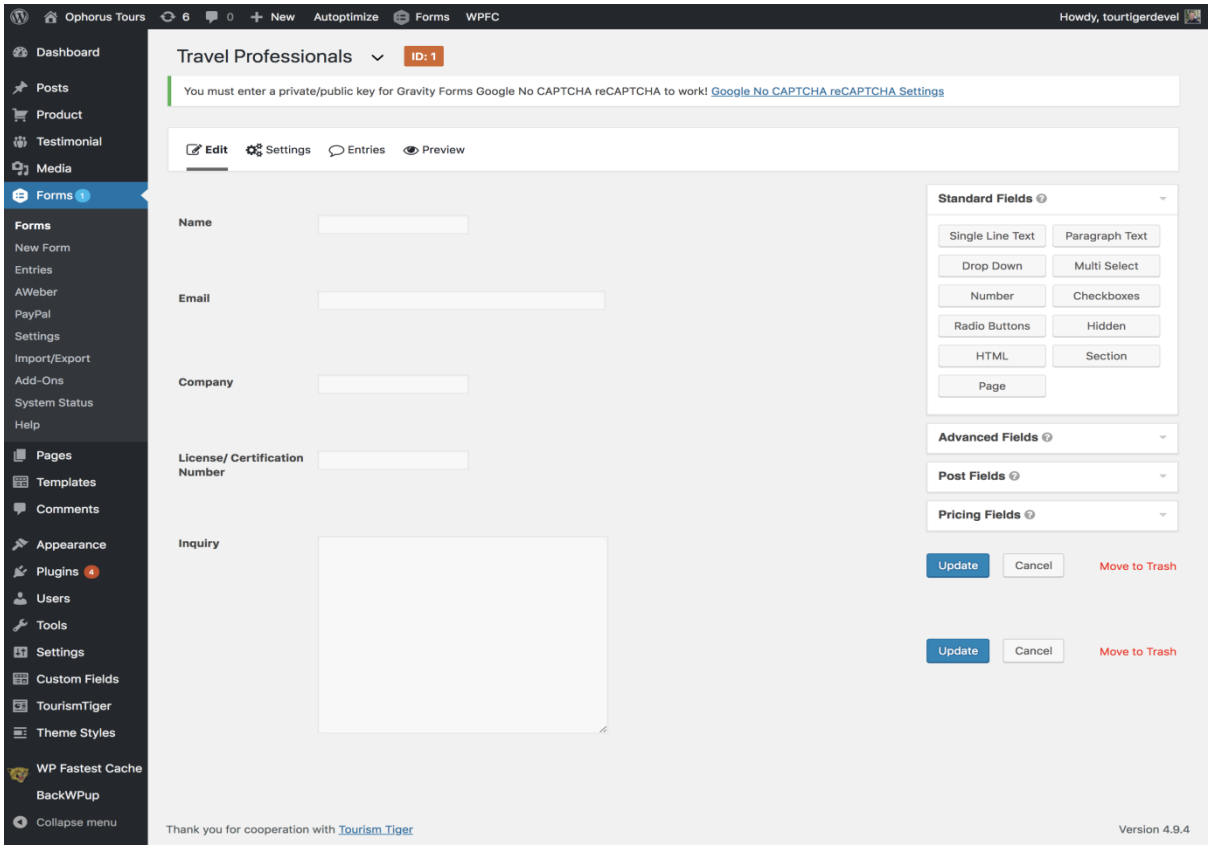


Рис. 31

1.4. Кешування статичних файлів веб-сайту

Autoptimize – плагін для стискання каскадних таблиць стилей, скриптів та html за допомогою можливостей мови програмування PHP.

Перше, що потрібно зробити для оптимізації швидкості завантаження сторінки - це зменшити кількість HTTP запитів. Один з методів зробити це в WordPress - це зібрати всі підключені скрипти та стилі в один. Така операція називається конкатенація (об'єднання, зчеплення).

Autoptimize об'єднує всі .css стилі на сторінці в один файл і поміщає його вгору документа. Так само об'єднує всі .js файли та вбудовані скрипти

(<script>) і розміщує їх у кінці документа. І є можливість сжадати саму HTML сторінку (видалити зайві пробіли і переноси строк).

Принцип роботи плагіна: перед тим, як вивести HTML код сторінки, плагін знаходить і вирізає всі .js файли, оптимізує і зменшує вилучені файли, записує їх у єдиний файл, зберігає файл на сервері і розміщує посилання на цей єдиний файл на самому кінці сторінки, пред тег </ body>. Аналогічна операція проводиться і з. Css-файлами, тільки розміщуються вони в початок сторінки.

Після інсталяції плагіну нічого не сжимається, всі функції потрібно увімкнути окремо.

1.5. Автоматизація початку процессу розробки

Аналізуючи результат автоматизації розробки шаблону для розробки веб-сайтів, необхідно підготувати інтерфейс для розгортання стартової архітектури проекту для її подальшого розвитку. Використаємо платформу для зберігання вихідного коду Github, тому що вона надає можливість використовувати шаблони репозиторіїв для старту розробки проекту залежно від його типу.

GitHub - це глобальна компанія, яка забезпечує хостинг для управління версіями програмного забезпечення за допомогою Git. Він забезпечує контроль доступу та декілька функцій співпраці, таких як відстеження помилок, запити функцій, управління завданнями та вікі-файли для кожного проекту.

Окрім хостингу вихідного коду, GitHub підтримує такі формати та функції:

- документація, включаючи автоматично виведені файли README у різних форматах файлів;
- відстеження випусків (включаючи запити щодо функцій) за допомогою міток, етапів, правонаступників та пошукової системи;
- GitHub Actions, що дозволяє будувати постійну інтеграцію та безперервне розгортання конвеєрів для тестування, випуску та розгортання програмного забезпечення без використання сторонніх веб-сайтів / платформ;
- здійснює історію, тож на будь-якому єтапі розробки є можливість переглянути історію змін того чи іншого файлу
- попередження про безпеку відомих загальних уразливостей та експозицій у різних пакетах.

На Рис. 32 зображенний зовнішній вигляд репозиторію у якому розміщується вихідний код шаблону проектування веб-сайтів. Нам необхідно мати можливість створювати нові репозиторії на базі його.

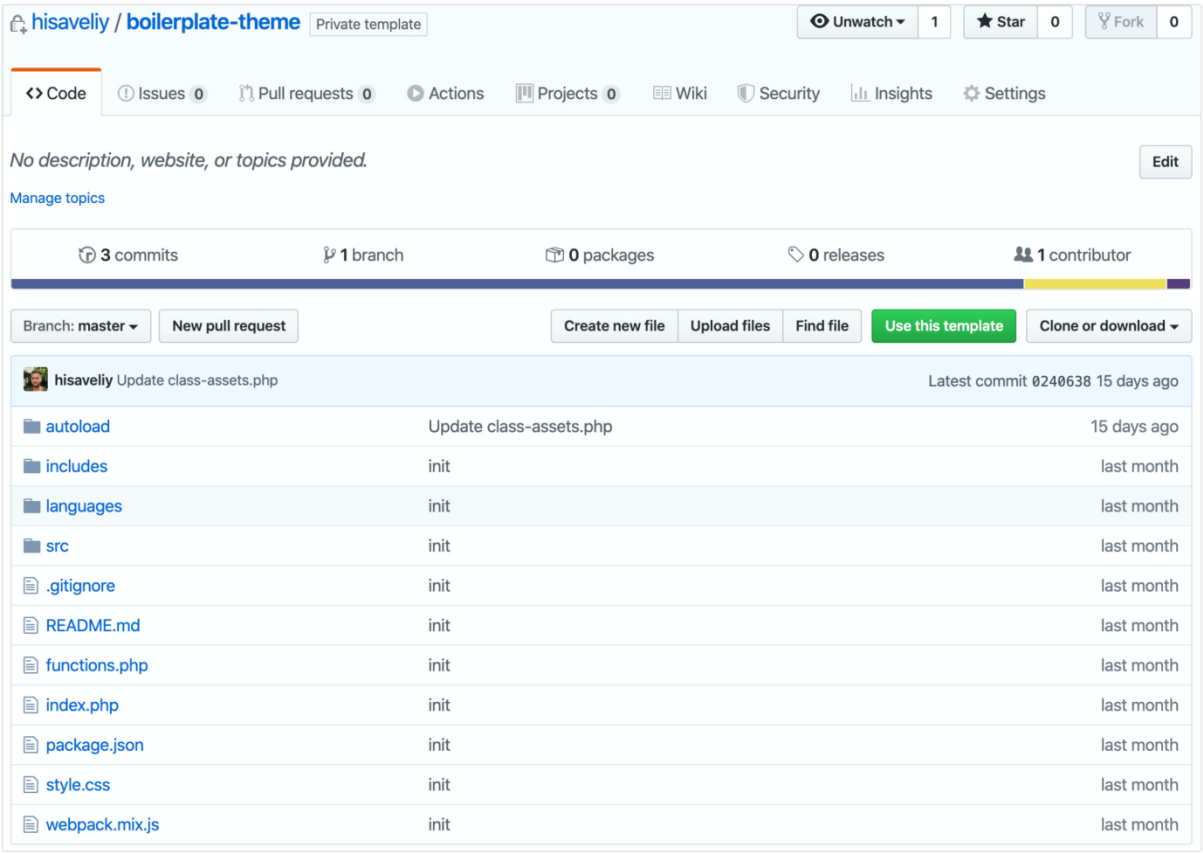


Рис. 32

Інструкція з використання коду

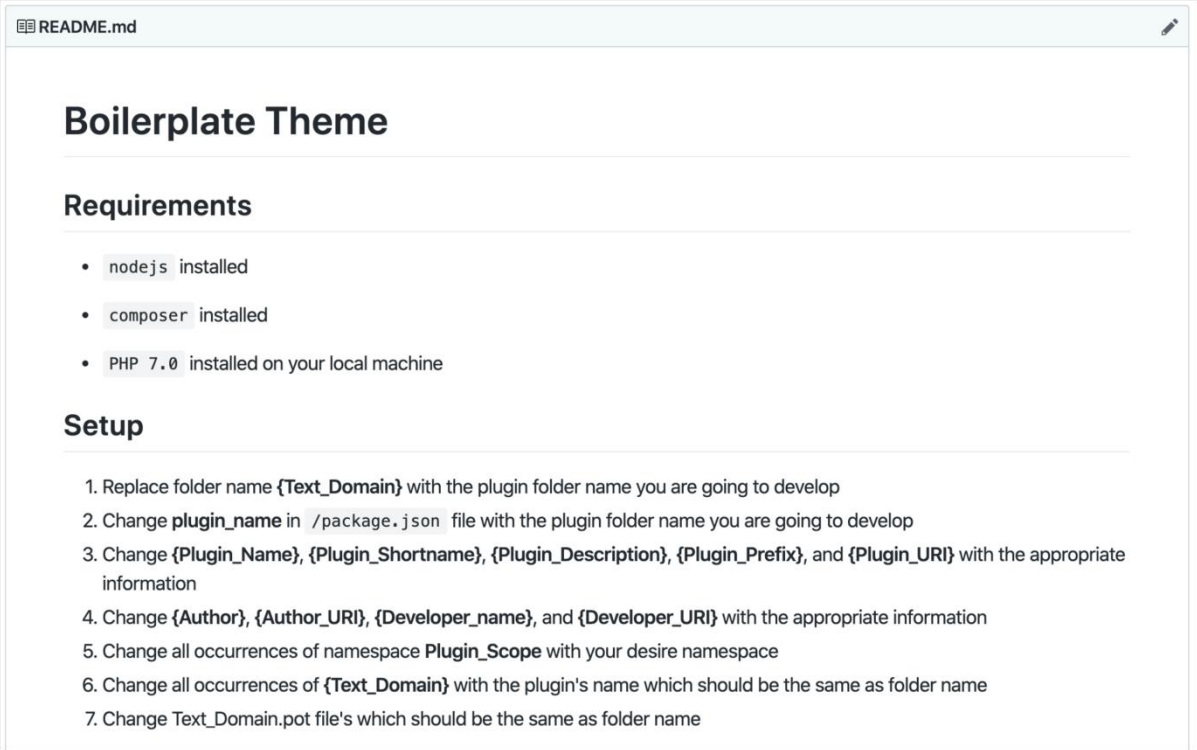
Для того щоб автоматизувати процес написання коду, необхідно зробити архітектуру зрозумілою для програмістів. В усіх без вийнятку проектах для вирішення цього питання використовують файл README.md який містить у собі інструкції для подальшого розвитку проекту.

README часто є першим пунктом, який відвідувач побачить під час відвідування сховища. Файли README зазвичай містять інформацію про:

- Що проект робить;
- Чому проект корисний;
- Як користувачі можуть розпочати роботу з проектом;

- Де користувачі можуть отримати допомогу щодо вашого проекту;
- Хто підтримує та сприяє проекту.

Інструкція до нашого шаблону містить також інформацію про те, які змінні необхідно замінини перед початком роботи над проектом. На Рис. 33 зображен фрагмент файлу README.md.


README.md

Boilerplate Theme

Requirements

- `nodejs` installed
- `composer` installed
- `PHP 7.0` installed on your local machine

Setup

1. Replace folder name **{Text_Domain}** with the plugin folder name you are going to develop
2. Change **plugin_name** in `/package.json` file with the plugin folder name you are going to develop
3. Change **{Plugin_Name}**, **{Plugin_Shortname}**, **{Plugin_Description}**, **{Plugin_Prefix}**, and **{Plugin_URI}** with the appropriate information
4. Change **{Author}**, **{Author_URI}**, **{Developer_name}**, and **{Developer_URI}** with the appropriate information
5. Change all occurrences of namespace **Plugin_Scope** with your desire namespace
6. Change all occurrences of **{Text_Domain}** with the plugin's name which should be the same as folder name
7. Change Text_Domain.pot file's which should be the same as folder name

Рис. 33

У документі згадується, що необхідно замінити назву веб-сайту, опис, та деякі змінні по всьому проекті.

Шаблон репозиторію

Розмістимо код теми у шаблон репозиторію, для його необхідно відвідати меню налаштувань репозиторію та відмітити чек-бокс «Template rrepositpry», розглянемо це на рис. 34.

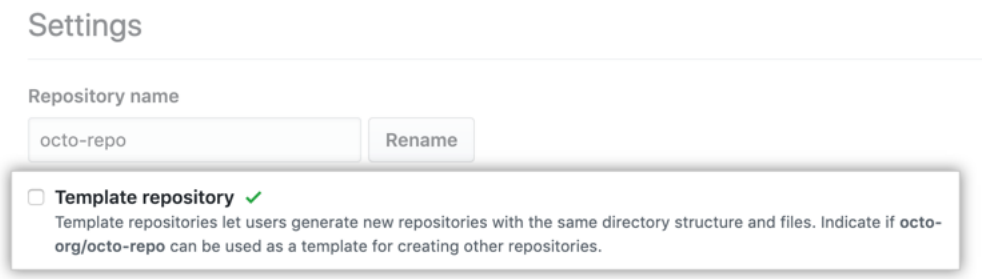


Рис. 34

Після того як репозиторій був зробленним шаблоном, кожен, хто має доступ до сховища, може створити нове сховище з тією ж структурою каталогу та файлами. Для цього необхідно виконати наступні кроки:

1. На GitHub перейти на головну сторінку сховища
2. Над списком файлів натиснути кнопку «Використовувати цей шаблон» як це зображено на Рис. 35

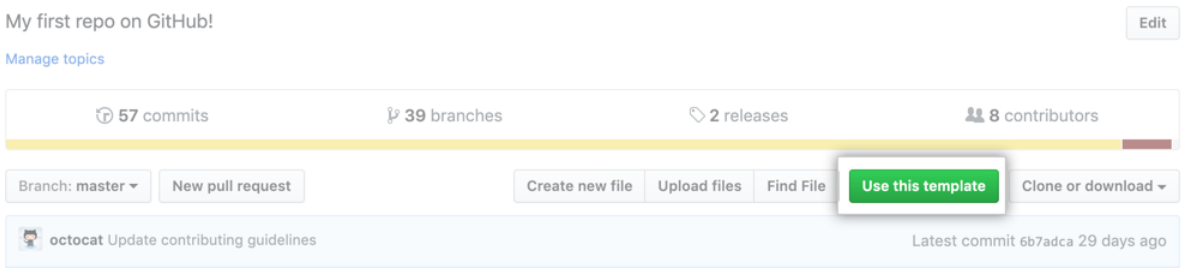


Рис. 35

3. Користуючись меню «Власник» необхідно обрати обліковий запис, який повінен володіти сховищем

4. Ввести назву сховища та необов'язковий опис

5. Необхідно обрати тип сховища зробивши його публічним чи приватним. Загальнодоступні сховища можуть бути видимими для загального користування, тоді як приватні сховища доступні лише їх власникам та людям, з якими був розділенний доступ

6. Для завершення цього процесу необхідно натиснути «Створити сховище з шаблону»

На Рис. 36 зображені налаштування сховища на момент його ініціалізації.

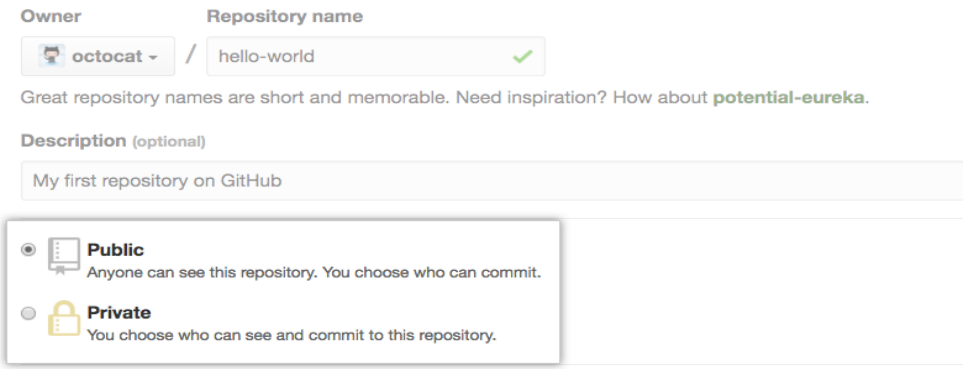


Рис. 36

Таким чином ми досягли того, що початок розробки веб-сайту автоматизований до мінімуму. У декільак кліків сховище коду, яке містить початковий функціонал, буде ініціалізованно.

ВИСНОВКИ

Ми автоматизували процеси проектування та розробкки веб-сайтів за допомогою написання шаблону на платформі Вордпресс, та використання інструментів планування та керування побудовою веб-сайтів Трелло. У висновку проаналізуємо автоматизовані етапи та веб-сайти побудовані за допомогою вихідного шаблону.

Етап проектування

Це етап автоматизованний за допомогою платформи Трелло, де шаблон дошки був розробленним, тож починаючи проектування нового веб-сайту, достатньо буде скопіювати шаблон у нову дошку, та почати додавання задач за допомогою шаблонів-задач відповідно до типу конкретної задачі.

Етап програмування

За допомогою використання фрейм-ворку Вордпресс, шаблону теми який було розробленно у роботі, та платформи хмарного зберігання коду Github, процес розробки зменшенний до мінімального набору під-етапів:

- розгортання git репозиторію на основі шаблону сховищ, який міститть у собі шаблон веб-сайту, та базову архітектуру;
- налаштування локального середовища для програмування, у тому числі, установлення платформи Вордпресс;
- програмування модулів та блоків інтерфейсу веб-сайту, яких бракує у шаблоні.

Етап наповнення контентом

Для зменьшення часу заповнення контенту на веб-сайтах, використовуется конструктор сторінок Гутенберг. У випадках коли функціоналу

конструктора сторінок недостатньо, маємо можливість до-програмувати необхідні блоки. Такі випадки мають бути спрогнозовані на етапі проектування, та реалізовані на етапі програмування.

Також, розробленно модуль стилізації веб-сайтів, де надано можливість корегувати зовнішній вигляд за допомогою зрозумілого інтерфейсу, таким чином, збережен час на персоналізацію веб-ресурсу.

Клієнтська частина

На рис. 37 представлені інтерфейси двох різних веб-ресурсів використовавших шаблон розроблений згідно з технологіями розібраними у дипломногу проекті.

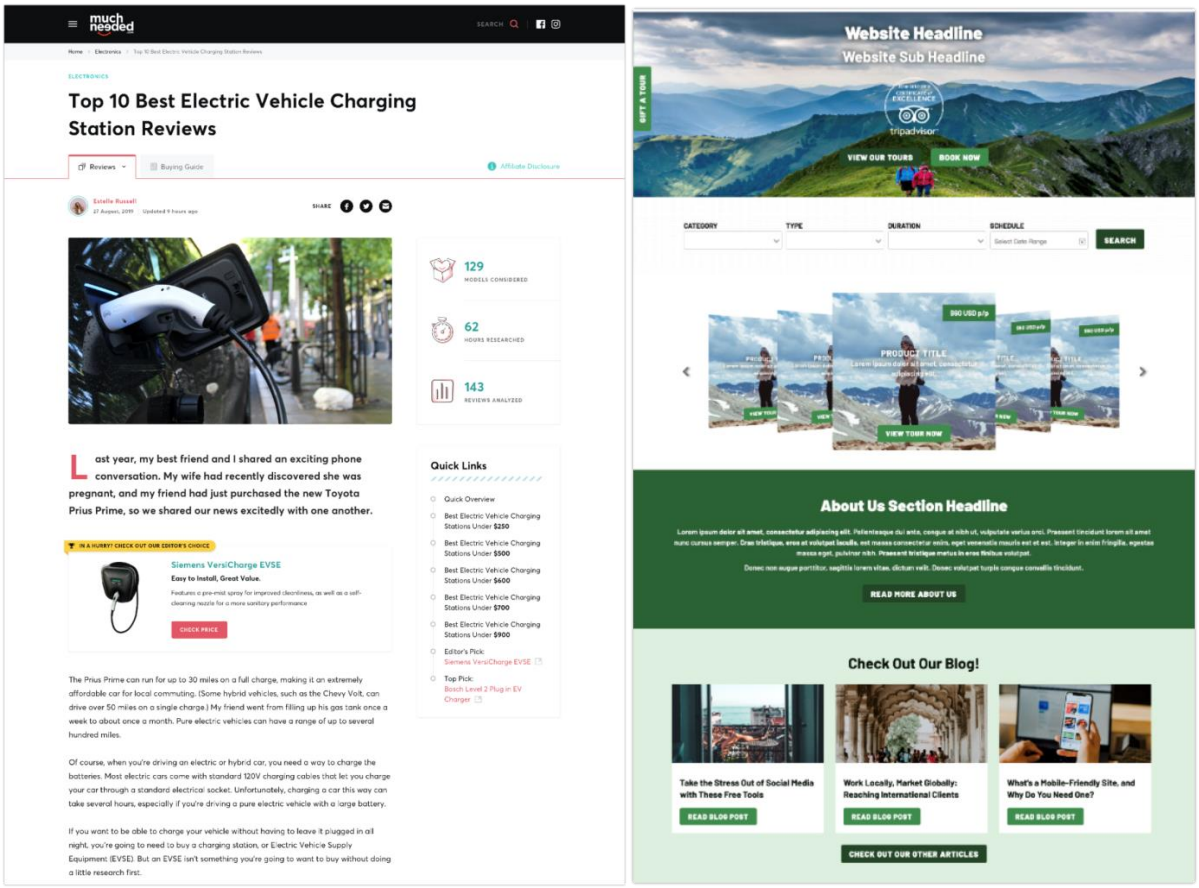


Рис. 37

Роблячи висновок із рис. 16, розроблена тема дозволяє доволі гнучко налаштувати інтерфейс кліентської частини веб-сайтів.

Управління контентом

На рис. 38 представлен вигляд редакторів різних сторінок. Деякі групи довільних полей вимушені були бути схованими, для того щоб зберегти якість зображеня.

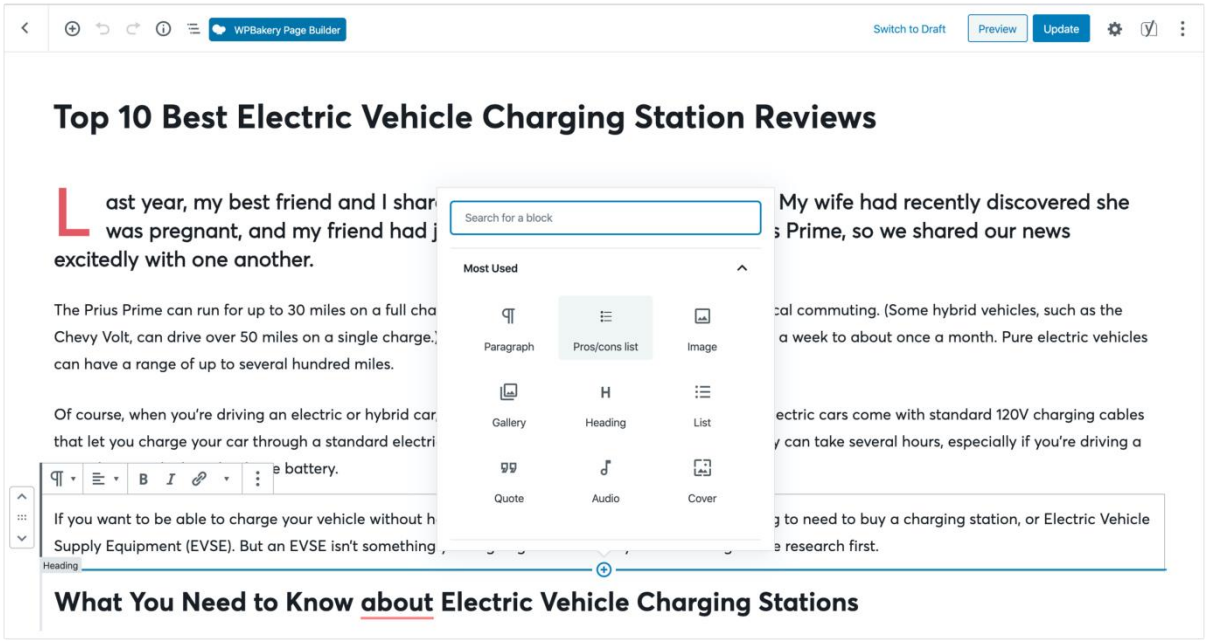


Рис. 38

Як видно на рис. 17, управління контентом не потребує уміння програмувати, хоча, має доволі гнучкі можливості конструювання сторінок.

Оптимізація

Посилаючись на розділ вступу до дипломної роботи, можна згадати твердження, що преміум шаблони які містять широкий набір функціоналу,

мають недолік у вигляді недостатньої оптимізації швидкості, тож честно буде поставити питання, чи достатньо оптимізованний шаблон, який було розробленно у рамках диплому?

Відповіддю на це питання буде звіт зробленний за допомогою платформи Gtmetrix (Рис. 39), у якому наведені характеристики оптимізації веб-сайту побудованного за допомогою використанних технологий.

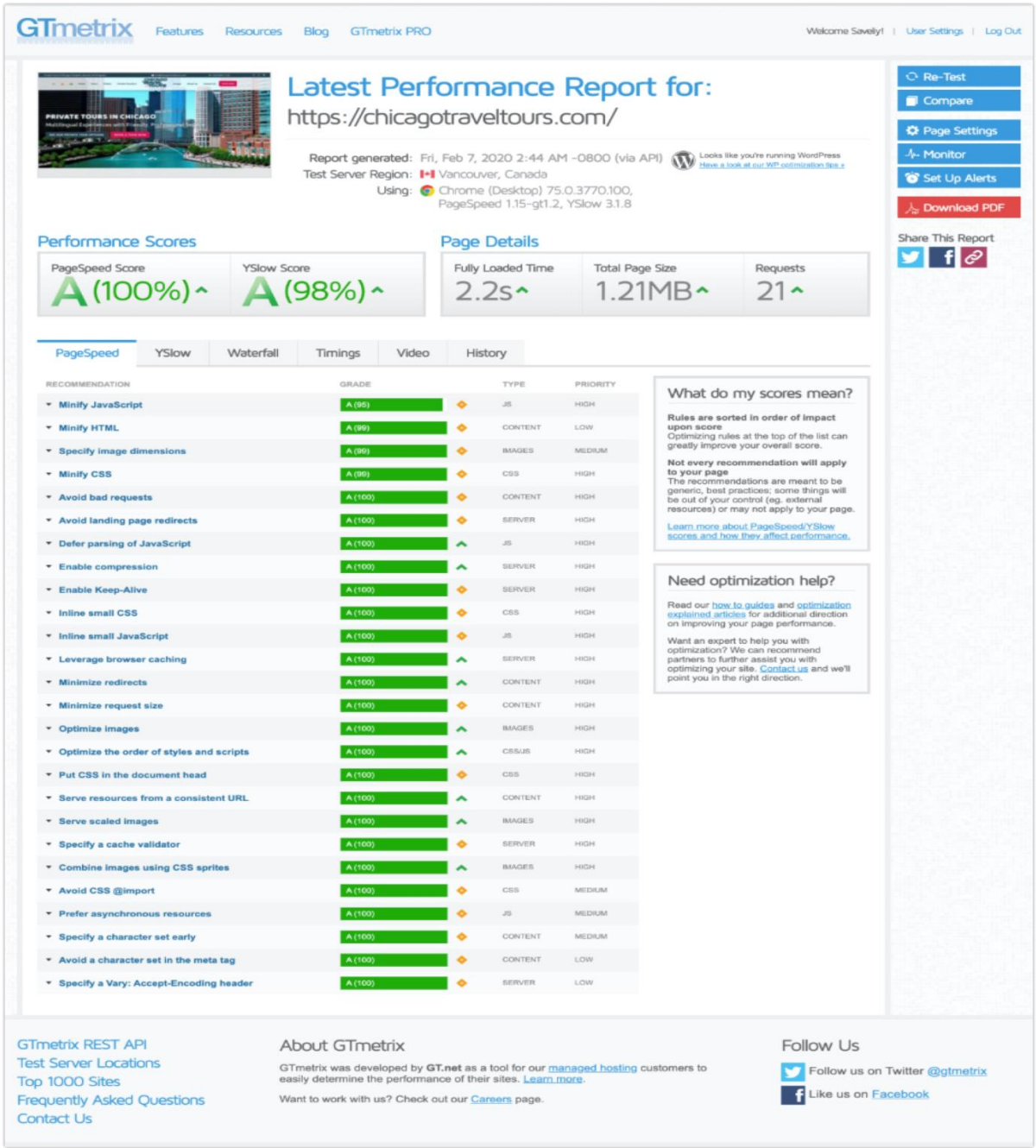


Рис. 39

Згідно з наведенив вище звітом, веб-сайт повністю оптимізовано, а результат повного завантаження сторінок у 2.2 секунди (середня швидкість будь-якого веб-сайту сягає семи секунд) свідчить, що веб-ресурс працьоє дуже швидко.

Тож, можна зробити висновок, що розгляненна технологія забезпечуе автоматизацію проектування та розробки веб-сайтів, проваджуючи можливість гнучко працювати із контентом на сайтах використовуючих систему управління контентом Ворпдресс, а також технологія надає увагу швидккості побудованних веб-сайтів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний веб-ресурс Asana. Режим доступу до веб-ресурсу: URL: https://asana.com/ вільний.
2. Офіційний веб-ресурс Trello. Режим доступу до веб-ресурсу: URL: https://trello.com/ вільний.
3. Офіційний веб-ресурс Jira. Режим доступу до веб-ресурсу: URL: https://www.atlassian.com/software/confluence вільний.
4. Офіційна документація Wordpress. Режим доступу до веб-ресурсу: URL: https://*wordpress.org*, вільний.
5. GIT --local-branching-on-the-cheap. Режим доступу до веб-ресурсу: URL: *https://git-scm.com*, вільний.
6. Хабрахабр. Режим доступу до веб-ресурсу: URL: habrahabr.ru, вільний.
7. ACF. Режим доступу до веб ресурсу: URL: *https://www.advancedcustomfields.com/*, вільний.
8. KeyCDN. Режим доступу до веб-ресурсу: URL: *https://www.keycdn.com/*, вільний.
9. ТОВ «Соціальний центр БАЛІ». Режим доступу до веб-ресурсу: URL: *http://scbali.com*, вільний.
10. Wikipedia. Режим доступу до веб-ресурсу: URL: https://uk.wikipedia.org/, вільний.
11. MAMP. Режим доступу до веб-ресурсу: URL: https://www.mamp.info/en/, вільний.
12. JetBrains. Режим доступу до веб-ресурсу: URL: https://www.jetbrains.com, вільний.
13. Stackoverflow. Режим доступу до веб-ресурсу: URL: https://stackoverflow.com, вільний.

14. “Learn JavaScript”, Режим доступу до веб-ресурсу: URL: https://learn.javascript.ru, вільний.
15. Gravity Forms. Режим доступу до веб-ресурсу: URL: https://www.gravityforms.com/, вільний.
16. WpNice. Режим доступу до веб-ресурсу: URL: http://wpnice.ru/, вільний.